uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería en Tecnologías de Telecomunicación
Curso académico 2016-2017

Trabajo Fin de Grado

Estudio de ataques DDoS basados en DNS

Pablo Pérez Gómez

Tutor

Daniel Díaz Sánchez

Universidad Carlos III de Madrid a 22 de septiembre de 2017

Resumen

A lo largo de este documento, se presentará lo que trata de ser un estudio acerca de los ataques de denegación de servicio basados en servidores DNS (Domain Name System). Con el fin de poner al lector en situación, en primer lugar, se llevará a cabo una introducción en la cual se revisa la situación actual en la que nos encontramos frente a este tipo de ataques, así como, se explicará brevemente en qué consisten y cuál es la finalidad que persiguen.

Posteriormente, se muestra la planificación llevada a cabo para conseguir lograr los objetivos propuestos, además de poner en situación acerca del marco legal y socio-económico que envuelve y afecta directamente al desarrollo del proyecto. Previo a la presentación de las pruebas realizadas y los resultados obtenidos, se lleva a cabo una descripción de las herramientas empleadas de tal forma que los lectores puedan hacer un seguimiento del estudio de una manera más sencilla y amena.

Adicionalmente, se muestran los diferentes diseños y prototipos elaborados, hasta que finalmente se llega al desarrollo final del entorno en el cual se llevarán a cabo las diferentes fases del estudio. Dichas fases se desarrollan de manera progresiva de tal forma que se consiga lograr una denegación de servicio para de esta manera observar su comportamiento. Finalmente, se expondrán las conclusiones obtenidas y se analizará si el estudio ha cumplido o no con los objetivos marcados al inicio de este. Valorando si el resultado obtenido es satisfactorio o no.

Abstract

Throughout this document, it will be displayed what tries to be a study about Denial of Service attacks based on DNS servers. In order to put the reader in situation, first of all, be held an introduction in which we review the current situation of this kind of attacks, as well as briefly explained in what consists and what is the purpose pursued by them.

Subsequently, it will be shown the planning carried out to achieve the different objectives proposed, in addition to study about the legal and socio-economic framework which surrounds and affects the development of the project. Preceding the results of the different tests carried out, a description of the different used tools is made so that readers can follow up the study in a simpler way.

In addition, the different designs and prototypes developed are shown, until we finally reach the final environment in which the different phases of the study will be carried out. These stages unfold progressively so that a denial of service attack is accomplished so that we can monitor their behaviour.

Finally, the conclusions reached will be displayed and it will be analysed whether the study has complied or not with the objectives set at the beginning of it. Assessing if the result is satisfactory or not.

Índice

Índice de figuras

Capítulo 1

1 Introducción

Actualmente vivimos una gran revolución tecnológica a nivel mundial, nuestros hábitos están experimentando una deriva hacia una completa conectividad con el mundo que nos rodea, hechos que hace unos años parecían impensables, hoy son un elemento cotidiano más de nuestras vidas, hecho que queda especialmente reflejado en lo que se refiere a la penetración de los teléfonos móviles en nuestra sociedad, dato en el cual España es líder mundial.

Según datos recogidos en el informe 'Mobile en España y en el Mundo' del año 2017 e impulsado por la consultora de estrategia digital Ditrendia (digital marketing trends):

- El 66% de la población mundial cuenta ya con teléfono móvil, siendo España el líder mundial del ranking en penetración, registrando un 88% de usuarios únicos. Siendo este, el dispositivo más usado entre los españoles para acceder a internet, alrededor de un 94,6%. [1]

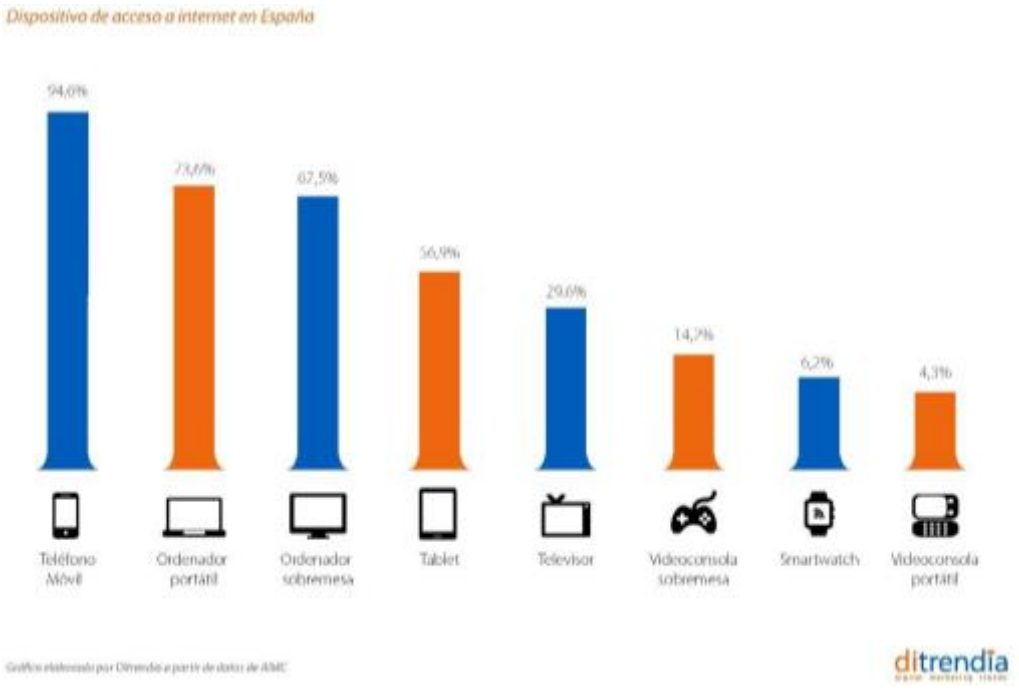


Figura 1: Dispositivos de acceso a Internet en España [1]

- La penetración de las tablets en nuestro país supera el 75%, si bien el tiempo dedicado a estas se encuentra en descenso desde 2015. [1]
- El Internet de las cosas, o según su acrónimo en inglés IoT (Internet of Things) es considerado un elemento muy importante para los directores de marketing, un 74% lo considera de gran importancia. [1]

- Derivado de esto, se prevé que el número de dispositivos conectados aumente un 23% al año hasta 2021 alcanzando de esta manera la cifra de 16.000 millones en todo el mundo y los 75.44 millones en 2025. [1]

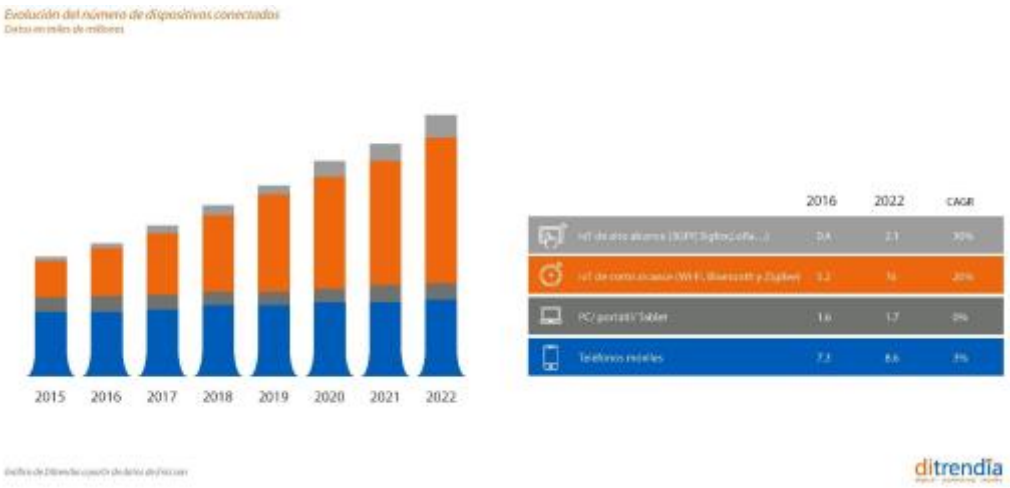


Figura 2: Evolución del número de dispositivos conectados [1]

- Como datos de interés que reflejan la importancia adquirida por estos dispositivos en nuestras vidas, en 2016, el 61% de los usuarios de teléfonos móviles aseguraba comprobar su dispositivo móvil en los primeros 5 minutos tras despertarse. Así como, el tiempo de media que un usuario hace uso de su móvil ronda los 170 minutos (2 horas y 50 minutos). [1]
- En España, el 92,8% de los usuarios hace uso de su móvil diariamente para acceder a internet, siendo un 37,7% quienes lo consideran el equipo principal para llevar a cabo esta tarea. Y un 99% de los jóvenes afirma acceder a diario a internet a través de su móvil. [1]
- En lo referente al comercio electrónico, el mobile commerce crece un 200% más rápido que el ecommerce, por lo que se espera que para finales de este año 2017 el 34% de las ventas online se hayan realizado a través de un teléfono móvil. [1]

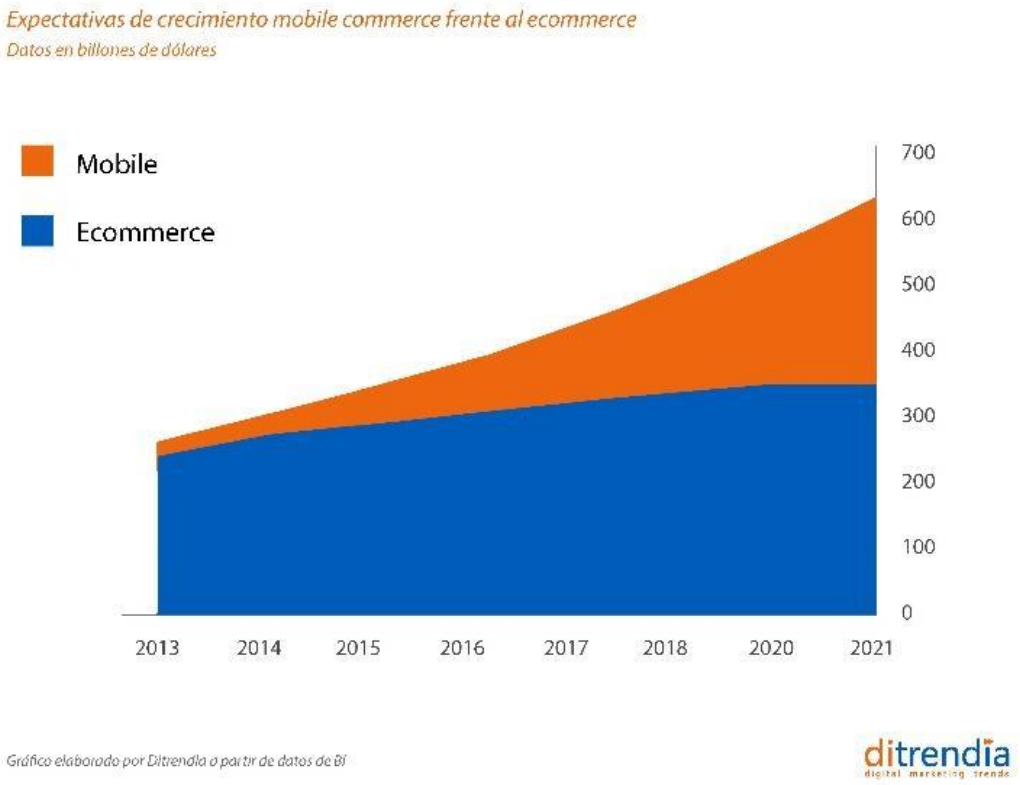


Figura 3: Expectativas de crecimiento mobile commerce frente al ecommerce [1]

- Como último dato, otro campo que se encuentra creciendo a gran ritmo es el de la banca móvil, en el cual el 25% de los usuarios ya solo emplea este método, siendo España junto con Holanda quienes poseen un mayor número de usuarios. [1]

Como dato de interés recogido en el informe 'Google Consumer Barometer Report' realizado por la compañía Mountain View con la colaboración de Kantar TNS, en 2012 la penetración de smartphones en España era de un 41%, si bien en tan solo cinco años, y en lo que a fecha del informe se refiere a inicios de este año, dicho valor se ha duplicado alcanzando el 81%. Este hecho refleja lo comentado previamente de cómo nuestra sociedad deriva a dicha completa conectividad con el mundo que nos rodea. [2]

Como consecuencia de este crecimiento de las tecnologías de comunicación, uno de los fenómenos que más ha crecido en estos años son los ataques de denegación de servicio o mayormente conocidos por sus siglas DDoS. Numerosos estudios reflejan el espectacular crecimiento de estos fenómenos, no ya solo en número de ataques sino también en su tamaño y duración.

Si nos remontamos al primer trimestre de 2016, basándonos en el Informe de Seguridad publicado por la empresa estadounidense Akamai, el número de ataques DDoS registrados por esta empresa respecto al mismo trimestre del año anterior (2015) aumentó en un 125%, la

mayoría de ellos con carácter de reflexión, principalmente sobre servidores DNS y NTP (Network Time Protocol). [3]

Situándonos ya en el primer trimestre de 2017, observamos que los ataques DDoS han incrementado tanto su frecuencia que han dejado de ser un hecho curioso para pasar a ser algo cotidiano, todo esto potenciado principalmente gracias a las botnets (redes de ordenadores zombis) generadas a partir del IoT, de seguridad muy baja, lo que provoca que en cada ataque se empleen cientos de miles de dispositivos. [4]

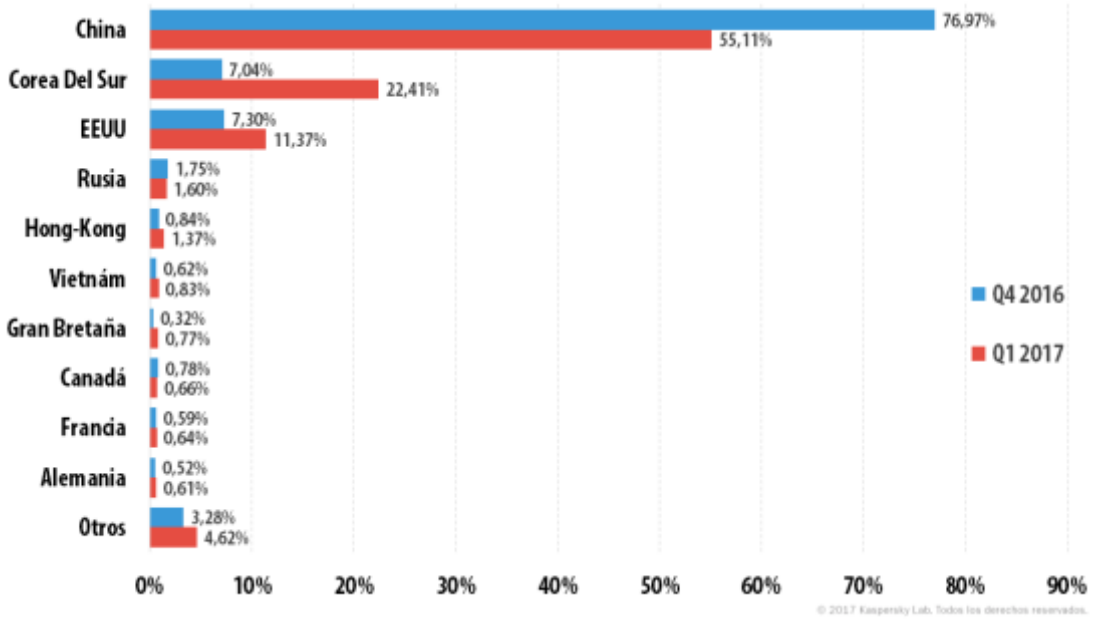


Figura 4: Distribución de ataques DDoS por país, último trimestre de 2016 y primero de 2017 [4]

Principalmente las motivaciones de los atacantes son el lucro directo, estos ataques suelen ser tan potentes que los objetivos afectados prefieren pagar el rescate a sufrir los daños tanto en infraestructuras como de reputación que estos suponen. Otra motivación que ha cobrado especial importancia es la causa política, especialmente en el segundo trimestre de este año 2017. [5]

Uno de las mayores fortalezas de estos ataques es la duración de estos, por ello los atacantes cada vez intentan llevar a cabo ataques más prolongados en el tiempo, y esto queda reflejado en que en este segundo trimestre de 2017 el ataque más largo registrado tuvo una duración de 277 horas, un 131% mayor que en el primer cuatrimestre. [5]

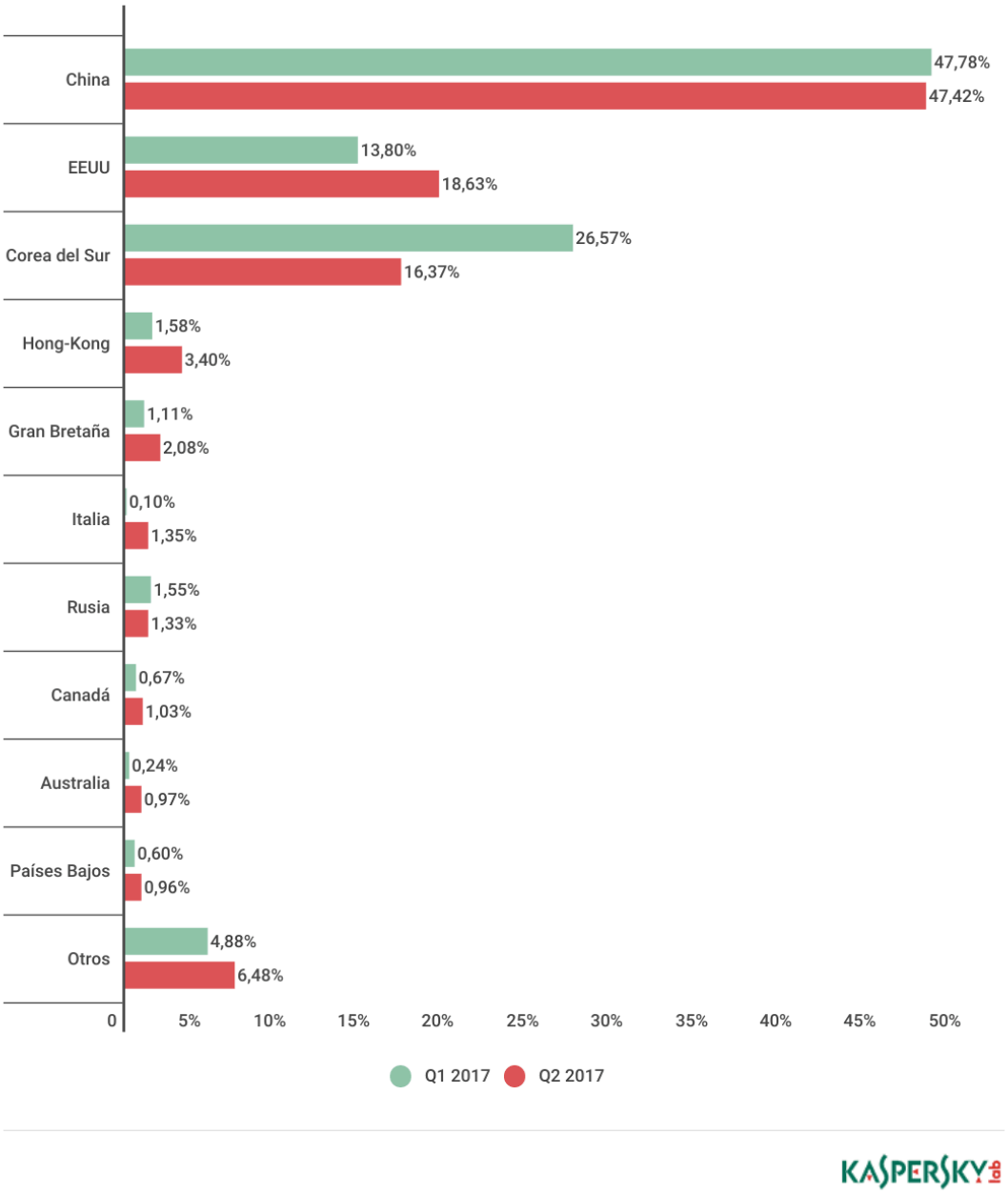


Figura 5: Distribución de ataques DDoS por país, primer y segundo trimestre de 2017 [5]

1.1 Ataques DoS y DDoS. Concepto.

En esta sección del documento abordaremos la definición de los conceptos de ataques DoS (Denial of Service o Denegación de Servicio) y DDoS (Distributed Denial of Service o Denegación de Servicio Distribuido), así como, la presentación de los tres grandes tipos de ataques, sin entrar en grandes detalles acerca de cada uno de ellos. Previamente, antes de abordar dicha clasificación, se ha de clarificar cual es el objetivo que se persigue llevando a cabo estos ataques, el cual es común en ambos casos, impedir el normal funcionamiento de la víctima receptora del ataque [6]. Tanto los ataques DoS como DDoS, se pueden dividir en tres grandes grupos:

- **Ataques volumétricos:** consiste en derivar un gran volumen de tráfico hacia la máquina de la víctima, dicho tráfico se encuentra constituido principalmente por paquetes UDP (User Datagram Protocol) e ICMP (Internet Control Message Protocol), los cuales con una pequeña cabecera son capaces de generar un gran volumen de tráfico, unido al hecho de que el procesado de estos paquetes no es trivial. Su magnitud se mide en bits por segundo (Bps). [7]

- **Ataques de protocolo:** dirigidos principalmente a las capas de sistema y de transporte, explotan las vulnerabilidades de los diferentes protocolos con el fin de consumir sus recursos asignados. También pueden ser dirigidos a servicios intermediarios como firewalls o balances de carga. Se miden en paquetes por segundo (Pps). [7]

- **Ataques de capa de aplicación:** dirigidos principalmente a servidores web, se emplean peticiones falsas enmascaradas, de manera que la víctima deje de dar servicio. La magnitud de medida es de peticiones por segundo (Rps). [7] [8]

Más adelante en el documento, se abordará en mayor detalle tanto la definición de los conceptos, así como de los diferentes tipos de ataques que hasta hoy día se conocen. No tanto abarcando el conjunto de ellos, si no centrándonos más específicamente en casos particulares.

1.2 Objetivos

Como en todo proceso de estudio o investigación, en primer lugar, hemos de fijarnos unos objetivos generales de gran tamaño, de esta manera podremos a posteriori fijar unos objetivos específicos los cuales podremos cumplir y alcanzar.

Los objetivos generales que se perseguían al inicio de este trabajo de fin de grado eran principalmente ahondar en toda aquella materia relacionada con los ataques de denegación de servicio y analizar tanto su forma de proceder como el efecto provocado en las victimas. Dicho objetivo viene motivado por la curiosidad surgida a lo largo de los estudios del grado, en los cuales tanto la terminología DoS como DDoS siempre se encontraban presentes, pero nunca se llegó a profundizar en ellos.

Como se ha podido comprobar en el apartado anterior, el campo de conocimiento que abarcan dichos ataques es muy amplio y extenso, cada uno de los diferentes ataques requiere un estudio detallado con el fin de conocer así, el verdadero funcionamiento interno de estos. Ya que esto no es posible debido al tiempo disponible para la realización del trabajo, se decidió seleccionar

específicamente una serie de objetivos concretos, centrándonos de esta manera en un tipo de ataque en particular.

Por ello, se decidió establecer como objetivos específicos el estudio y desarrollo en un entorno controlado de un ataque de tipo amplificación mediante servidores de DNS. Dentro de ese objetivo específico, se fijaron unas pautas de procedimiento, en las cuales se organizaba la manera de llevar a cabo el estudio. Se fijaron tres hitos a partir de los cuales se determinaba el nivel de desarrollo del trabajo: el primero, era la creación del entorno controlado y de los elementos necesarios; el segundo, consistía en llevar a cabo el ataque, probando diferentes escenarios y número de elementos; y el tercero e hito final y por tanto objetivo principal del estudio, es el análisis de los resultados obtenidos en las diferentes pruebas y observar bajo qué circunstancias se logra llevar a cabo la denegación de servicio.

Capítulo 2

2 Historia del proyecto

2.1 Planificación

Todo proyecto requiere de una planificación previa con el fin de tener organizado todo el proceso de desarrollo y de esta manera poder siempre encontrarnos en los plazos previstos para su realización. De esta manera, se decidió fijar unos límites temporales para las diferentes fases del estudio.

La duración total del proyecto se fijó en un periodo 5 meses y medio, contando que cada mes tendría una media de 4 semanas esto nos da un plazo de 22 semanas para su realización. Una vez fijada la duración total, se llevó a cabo la estructuración por fases:

- En primer lugar, la lectura de la documentación e instalación de las herramientas necesarias para desarrollar el estudio. Para llevar a cabo estas tareas, se fijó un plazo de mes y medio, es decir, seis semanas.
- Posteriormente, se ha de crear el entorno sobre el cual se realizarían las pruebas, esto es, la instalación de los diferentes elementos internos que componen nuestra red, y verificar el correcto funcionamiento de estos. El periodo de tiempo fijado es de dos meses, es decir, ocho semanas.
- Para la realización de las pruebas previas al estudio en sí, en las cuales se analiza de manera superficial el comportamiento de los diferentes elementos, observando si existe alguna anomalía de carácter reseñable, se fijó un plazo de dos semanas.
- Finalmente, las seis semanas restantes, se encuentran destinadas a la redacción del documento final, así como a la realización y análisis en detalle de las diferentes pruebas del estudio.

De esta manera se obtienen las 22 semanas totales, divididas en cuatro fases diferentes, destinadas a cubrir el total desarrollo del proyecto. Esta planificación nos será útil más adelante, ya que a partir de ella podremos fijar un presupuesto válido para la realización del proyecto.

2.2 Marco regulador

A lo largo de esta sección se llevará a cabo un análisis del marco regulador de la ciberseguridad en España. En primer lugar, se presentará dicho término y lo que actualmente se conoce como seguridad en la red, seguido de un análisis de los delitos más recurrentes y frecuentes, todo ello

complementado con el conjunto de leyes y normas que lo regulan y las cuales establecen la barrera entre la legalidad y la ilegalidad. Por último, y para finalizar con este apartado, se nombrarán diferentes instituciones encargadas de velar por el cumplimiento de estas normas.

2.2.1 Ciberseguridad

Para la definición del término ciberseguridad se recurre al artículo 2.3 de la Orden Ministerial 10/2013, de 19 de febrero por la que se crea el Mando Conjunto de la Ciberdefensa, publicada en el Boletín Oficial del Ministerio de Defensa [9]. En este documento se expone dicho término como el conjunto de actividades dirigidas a proteger el ciberespacio contra el uso indebido del mismo, defendiendo su infraestructura tecnológica, los servicios que prestan y la información que manejan. [10]

De manera más sencilla, ciberseguridad engloba todos aquellos mecanismos de seguridad en Internet con el fin de la protección de la información del usuario, ya sean datos personales, información bancaria, compras online…

Con el fin de regular todo lo relacionado con la ciberseguridad, y lo que esta supone, dentro del BOE (Boletín Oficial del Estado) se incluye en el apartado de códigos electrónicos lo que se conoce como el Código de Derecho de la Ciberseguridad (edición actualizada a 9 de marzo de 2017). En este, se encuentran recogidas todas aquellas leyes y reales decretos que determinan el procedimiento de actuación ante determinadas situaciones, todas ellas relacionadas con la seguridad en la red. La normativa de Seguridad Nacional, el procedimiento de respuesta a incidentes de seguridad, protección de infraestructuras críticas, ciberdelincuencia, etc… todos ellos son diferentes apartados recogidos en dicho documento. Hacer especial mención al apartado de ciberdelincuencia, el cual recoge la siguiente legislación [11]:

- Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal. [11]
- Ley Orgánica 5/2000, de 12 de enero, reguladora de la responsabilidad penal de los menores. [11]
- Real decreto de 14 de septiembre de 1882 por el que se aprueba la Ley de Enjuiciamiento Criminal. [11]

2.2.2 Ciberdelincuencia

Podemos definir el término ciberdelincuencia como cualquier tipo de actividad delictiva que implica el uso de equipos informáticos o de internet, normalmente con el objetivo de obtener información privada, realización de actividades fraudulentas, dañar o destruir otros equipos o redes de comunicación, pornografía infantil, etc… [12]

Para la clasificación de estos delitos, haremos uso de la clasificación llevada a cabo por la Brigada de Investigación Tecnológica de la Policía Nacional Española [13], la cual es la destinada a responder a los retos que plantean las nuevas formas de delincuencia [14]:

- **Ataques contra el derecho a la intimidad:** constituyen aquellos delitos de descubrimiento y revelación de secretos mediante el apoderamiento y difusión de reservados registrados en ficheros o soportes informáticos. (Artículos del 197 al 201 del Código Penal) [13]
- **Infracciones a la Propiedad Intelectual a través de la protección de los derechos de autor:** especialmente la copia y distribución no autorizada de programas de ordenador y tenencia de medios para suprimir los dispositivos utilizados para proteger dichos programas. (Artículos 270 y otros del Código Penal) [13]
- **Falsedades:** concepto de documento como todo soporte material que exprese o incorpore datos. Extensión de la falsificación de moneda a las tarjetas de crédito y débito. Fabricación o tenencia de programas de ordenador para la comisión de delitos de falsedad. (Artículos 386 y ss. del Código Penal) [13]
- **Sabotajes informáticos:** delito de daños mediante la destrucción o alteración de datos, programas o documentos electrónicos contenidos en redes o sistemas informáticos. (Artículo 263 y otros del Código Penal) [13]
- **Fraudes Informáticos**: delitos de estafa a través de la manipulación de datos o programas para la obtención de un lucro ilícito. (Artículos 248 y ss. del Código Penal) [13]
- **Amenazas:** realizadas por cualquier medio de comunicación. (Artículos 169 y ss. Del Código Penal) [13]
- **Calumnias e injurias:** cuando se propaguen por cualquier medio de eficacia semejante a la imprenta o la radiodifusión. (Artículos 205 y ss. Del Código Penal) [13]
- **Pornografía infantil:** entre los delitos relativos a la prostitución al utilizar menores o incapaces con fines exhibicionistas o pornográficos. [13]
    - La inducción, promoción, favorecimiento o facilitamiento de la prostitución de una persona menor de edad o incapaz. (Art. 187)
    - La producción, venta, distribución, exhibición, por cualquier medio, de material pornográfico en cuya elaboración hayan sido utilizados menores de edad o incapaces, aunque el material tuviere origen en el extranjero o fuere desconocido. (Art. 189)
    - El facilitamiento de las conductas anteriores. (Art. 189)

- La posesión de dicho material para la realización de dichas conductas. (Art. 189)

A continuación, presentaremos cuales son los ciberdelitos más comunes y en que consiste cada uno de ellos, dejando un poco de lado el marco legal y centrándonos en el aspecto práctico de estos:

- **Phising / Spoofing:** consiste en acceder ilegalmente a un equipo informático y desde este enviar múltiples correos electrónicos; falsificar información del encabezado en varios mensajes de correo electrónico; o enviar varios mensajes de correo electrónico comercial con la intención de engañar a los destinatarios. [15]
- **Extorsión:** hacer uso de internet para amenazar con la intención de extorsionar a un individuo para conseguir dinero u otra cosa de valor. [15]
- **Hacking:** acceder de forma ilegal a datos almacenados en un ordenador o servidor [15]. Un caso bastante relevante fue la filtración de cerca de 165 millones de cuentas de LinkedIn en 2012, si bien no se supo de dicha filtración hasta 2016. [16]
- **Apuestas ilegales:** participar a través de internet en el negocio de las apuestas ilegales, en cualquier evento deportivo o concurso. [15]
- **Fraude:** defraudar, así como, obtener dinero o bienes mediante pretextos falsos, todo ello de manera premeditada y haciendo uso de internet para llevarlo a cabo. Previamente también se ha definido como delitos de estafa a través de la manipulación de datos o programas para la obtención de un lucro ilícito. [15]
- **Ciberbulling:** consiste en acosar, amenazar o abusar de una persona haciendo uso de internet, en la mayor parte de los casos se lleva a cabo de manera anónima. Este es uno de los delitos que más ha crecido en los últimos años junto con el fraude. [15]
- **Pornografía infantil:** como previamente se ha expuesto, consiste en utilizar menores o incapaces con fines exhibicionistas o pornográficos. [15]
- **Tráfico de drogas:** entrega, distribución o dispensación de sustancias ilegales o medicamentos que requieran de receta, haciendo uso de la red. [15]
- **Piratería:** infringir los derechos de autor para obtener ganancias financieras o distribuir sin permiso un trabajo que no es de tu propiedad intelectual. [15]

2.2.3 Instituciones implicadas

En este apartado presentaremos algunas de las instituciones implicadas en el campo de la ciberseguridad, ya sea estableciendo normas de regulación, o velando por el mantenimiento del orden y la seguridad en la red.

2.2.3.1 INCIBE

Anteriormente conocido como Instituto Nacional de Tecnologías de la Comunicación, S.A. (INTECO), en 2104 pasó a llamarse Instituto Nacional de Ciberseguridad de España, S.A. (INCIBE), según acuerdo adoptado en la Junta General celebrada el 27 de octubre de ese mismo año.

INCIBE es una institución dependiente del Ministerio de Energía, Turismo y Agenda Digital (MINETAD) a través de la Secretaría de Estado y para la Sociedad de la Información y Agenda Digital (SESIAD), siendo la entidad de referencia a nivel nacional para el desarrollo en ciberseguridad. INCIBE trata de desarrollar la ciberseguridad como motor de transformación social y oportunidad para la innovación, con una actividad basada en la investigación, la prestación de servicios y la coordinación con los agentes con competencias en la materia, liderando diferentes actuaciones a nivel nacional e internacional. [17]

2.2.3.2 CERTSI

El CERT (Computer Emergency Response Team) de Seguridad e Industria (CERTSI), es el centro de respuesta a incidentes de seguridad de la información del Ministerio de Energía, Turismo y Agenda Digital y del Ministerio del Interior. Gracias al acuerdo del Consejo Nacional de Ciberseguridad de 29 de mayo de 2015, el CERTSI se convierte en el órgano competente en la prevención, mitigación y respuesta ante incidentes cibernéticos. [18]

Se encuentra operado técnicamente por INCIBE y bajo la coordinación conjunta del CNPIC (Centro Nacional de Protección de Infraestructuras y Ciberseguridad) e INCIBE, el CERTSI fue constituido en 2012 bajo el Acuerdo Marco de Colaboración en materia de Ciberseguridad entre la Secretaria de Estado se Seguridad y la Secretaria de Estado de Telecomunicaciones y para la Sociedad de la Información. Actualmente se encuentra regulado por el Acuerdo de 21 de octubre de 2015, suscrito por ambas Secretarías de Estado.

2.2.3.3 CNPIC

El CNPIC se encarga de impulsar, coordinar y supervisar todas aquellas actividades que tiene delegadas la Secretaría de Estado de Seguridad del Ministerio de Interior en lo que se refiere a la protección de las infraestructuras críticas nacionales. Su objetivo principal consiste en poder garantizar la seguridad de las infraestructuras que proporcionan los servicios básicos e indispensables a la sociedad. Estas acciones pasan por optimizar la seguridad de dichas infraestructuras, las cuales se enmarcan principalmente en el ámbito contra agresiones deliberadas y especialmente, contra ataques terroristas. [19]

2.2.3.4 GDT

Perteneciente a la Guardia Civil, el Grupo de Delitos Telemáticos se encuentra englobado en la Unidad Central Operativa, y es el encargado de investigar todos aquellos delitos que se lleven a cabo a través de la red. El origen de esta unidad tiene lugar en 1996, y ha ido evolucionando según las necesidades que han ido surgiendo en la red, hasta derivar en lo que es hoy en día. Su trabajo se encuentra apoyado también en los Equipos de Investigación Tecnológica (EDITE,s), los cuales se encuentran presentes en cada una de las provincias de España.

Cabe destacar su membresía en los Grupos de Trabajo de Interpol en Europa y Latinoamérica, en el Foro Internacional del G-8 para el cibercrimen, y en Grupo de Europol. [20]

2.3 Contexto socio-económico

Como en todo proyecto, se ha de valorar el contexto social y económico en el cual nos encontramos, lo cual pasa por establecer un presupuesto para el desarrollo del estudio, y evaluar el impacto socio-económico que este tendrá. De esta manera podemos determinar si nuestra investigación es rentable o no, tanto económicamente como a lo que a la sociedad se refiere.

2.3.1 Presupuesto

Para la elaboración del presupuesto se tuvieron en cuenta los tres factores principales presentes en el proyecto: un equipo informático en el cual llevar a cabo las pruebas del estudio; las horas de trabajo propias como investigador; y finalmente, las horas de consultoría externa proporcionadas por el tutor del proyecto.

- En cuanto al equipo informático, se optó por el uso de un MacBook Pro, el cual tiene un coste de 1.500€.
- Referente a lo que supone el trabajo por horas como investigador, se establece un precio por hora de 25 €, y teniendo en cuenta el número de horas trabajadas, las cuales son 35 horas semanales de un total de 22 semanas, el montante total es de 19.250€. [21]
- Por último, el precio de la consultoría externa se establece en 90 €/hora, por un total de 24 horas, se obtiene un coste de 1.680€. [22]

Tabla 1: Presupuesto proyecto

Concepto	Precio/Hora	Nº horas	Total
Investigación	25 €/hora	770 horas	19.250 €
Consultoría externa	90 €/hora	24 horas	2.160 €
MacBook Pro	----------	----------	1.500 €
TOTAL	----------	----------	**22.910 €**

Por tanto, el montante total del presupuesto del proyecto es de 22.910€.

2.3.2 Impacto socio-económico

El impacto social y económico del proyecto viene dado principalmente por el gran crecimiento que han experimentado los ataques de denegación de servicio en estos últimos años, tal como se ha visto expuesto en la presentación del trabajo. Tal ha sido el crecimiento de estos, que han llegado a afectar de alguna manera, ya sea directa o indirecta, nuestras vidas.

El pasado 21 de octubre de 2016, la empresa Dyn, un importante proveedor de servicios DNS, fue víctima de un ataque de denegación de servicio, el cual afectó a grandes plataformas como Spotify, Twitter, AirBnB o Paypal entre otros, lo que provocó que experimentaran caídas parciales o totales durante varias horas, especialmente en la costa este de Estados Unidos [23]. Con esto, se pretende ilustrar cómo estos ataques pueden influir en nuestras vidas, ya que las plataformas anteriormente mencionadas se encuentran presentes de manera asidua en nuestras vidas, y por tanto estos ataques pueden llegar a provocar una alteración en el normal funcionamiento de estas.

Por ello, desde el aspecto social, la finalidad del proyecto es entender el funcionamiento de este tipo de ataques, documentándolo de manera detallada de tal forma que el lector pueda comprender en qué consisten y cómo se desarrollan. De esta manera, y con una sociedad más concienciada en este aspecto, se podrían aumentar las medidas de seguridad en los dispositivos personales, evitando de esta manera la fácil creación de botnets que deriven en un posterior ataque de denegación de servicio.

En lo referente al impacto económico, como previamente se ha expuesto en la parte final de la introducción, una de las principales motivaciones de este tipo de ataques es el lucro directo, la víctima mediante el pago del rescate evita la realización del ataque, y por tanto las pérdidas

económicas que este le pueda producir debido a daños en infraestructuras o pérdida de reputación.

Extrapolando esto a un nivel de usuario, hoy en día con el gran acceso a todo tipo de información que se tiene, cualquier usuario podría ser capaz de aprender cómo realizar un ataque a pequeña escala, por ejemplo, a una pequeña página web, la cual no cuente con unos servidores de gran capacidad. Además, no hemos de olvidarnos del mercado que existe alrededor de ellos, ya que en el mercado negro podemos encontrar hackers que ofrecen sus servicios para este tipo de fines, con un precio medio de 18 €/hora, según un informe de la empresa Kaspersky Lab [24].

Por ello, hemos de ser conscientes de que esto puede derivar en una espiral de continuos ataques ya no solo a grandes corporaciones, sino a pequeñas y medianas empresas, como consecuencia de competidores directos, que deseen llevar a cabo una competencia desleal, generando una mala experiencia al usuario al tratar de acceder a los servicios o información de dichas empresas.

Para finalizar, remarcar la importancia del impacto socio-económico del proyecto, ya que, en un futuro no muy lejano, los ataques de denegación de servicio podrían llegar a ser un elemento cotidiano en nuestras vidas, y por tanto hemos de conocerlos y saber cómo funcionan, de tal forma que podamos minimizar sus consecuencias.

Capítulo 3

3 Estado del arte

3.1 Ataques DoS y DDoS. Diferencias y tipos

Anteriormente en la introducción, se llevó a cabo una pequeña presentación de lo que son y en consiste llevar a cabo ataques tanto DoS como DDoS. A lo largo de esta sección, se llevará a cabo una profundización en estos términos, explicando de manera más detallada las diferencias entre ambos, así como, mostrando diferentes tipos de ataques, esta vez no genéricos, si no ya casos particulares englobados dentro de los tres grandes grupos vistos previamente.

3.1.1 Ataques DoS

Los ataques de denegación de servicio o ataques DoS consisten principalmente en impedir el normal funcionamiento de la máquina de la víctima del ataque, de esta manera se consigue como su propio nombre indica, una denegación del servicio. La principal característica a tener en cuenta, y la cual la diferencia en su totalidad respecto a los ataques DDoS, es que estos ataques se llevan a cabo desde una única máquina, siendo su principal objetivo inundar a la víctima a través de numerosas peticiones, o directamente, provocar su colapso [25] [26]. Principalmente suelen desarrollarse a nivel local [27]. Existen diferentes tipos de ataques DoS, los cuales se procede a explicar a continuación:

- **Ataque de amplificación mediante DNS:** consiste principalmente en generar peticiones falsas a un servidor DNS, dichas peticiones parecen ser creadas por la víctima, y por tanto la respuesta de la ya mencionada petición será enviada al receptor del ataque. El termino amplificación, viene dado ya que una vez se genera dicha petición al servidor DNS, la respuesta que este emite es de un tamaño mucho mayor, generando así, la denegación de servicio en la víctima. [25]

- **Ataques en la capa de aplicación:** en esta ocasión el atacante busca generar una gran cantidad de trafico ilegitimo a servidores de aplicación en internet, principalmente del tipo HTTP (Hypertext Transfer Protocol) o DNS. Llegados a este punto, dependiendo de la complejidad del ataque, podemos encontrarnos con ataques de un carácter más sencillo, cuyo único objetivo es buscar la inundación de dichos servidores; o ataques de carácter más complejo, los cuales ya tratan de buscar vulnerabilidades en dichos servidores o directamente en el protocolo empleado por la víctima. [25]

- **Ataques por inundación de buffer:** dicho ataque es el más común en cuanto a lo que ataques DoS se refiere, consiste principalmente en enviar una cantidad de tráfico de datos mucho mayor que la que la maquina pueda procesar por diseño. Algunos de ellos, se encuentran especialmente diseñados para explotar debilidades en diferentes aplicaciones o servidores. [25] [26]

- **Ataques ping-of-death:** consiste principalmente en abusar de un simple comando de consola conocido como ping (Packet Inter-Network Groper). De esta manera, el atacante se aprovecha del protocolo mediante el envío de solicitudes con un payload de gran tamaño (superior a 64KB), de esta manera se consigue la víctima se vea colapsada, evitando así, que responda a las solicitudes no maliciosas. [25] [27]

- **Ataques por inundación de peticiones SYN:** fundamentado en el protocolo TCP (Transmission Control Protocol), explota la posibilidad que ofrece el establecimiento de conexión entre dos máquinas (threeway handshake). El atacante envía un gran volumen de peticiones para establecer conexión, sin la intención de completar ninguna, de esta manera la víctima al no recibir la confirmación del establecimiento de conexión, prosigue enviando dichas respuestas a las peticiones SYN iniciales, consiguiendo de esta manera la denegación de servicio. Cabe destacar que la generación de un flujo de peticiones SYN tiene un coste relativamente bajo comparado con los recursos que se invierten para contestar a dichas peticiones. [25]

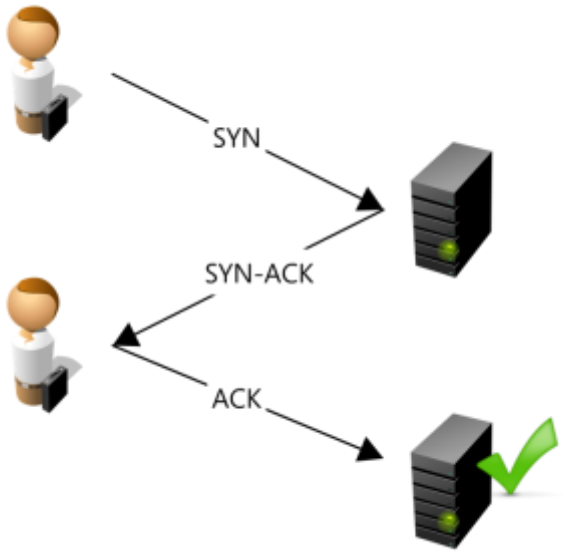


Figura 6: TCP Threeway Handshake [28]

- **Ataques por agotamiento (exhaustion attacks):** también conocidos como ataques TCP, el atacante modifica las tablas de estado contenidas ya sea en el firewall, routers o demás elementos, introduciendo en estas, datos relevantes para llevar a cabo el ataque. De esta manera, internamente el atacante puede ser capaz de abrir un número de conexiones mayor al que la víctima puede soportar. [25]

- **Ataque de la lágrima (teardrop attack):** explota vulnerabilidades en la manera en la cual antiguos sistemas operativos procesaban los paquetes IP (Internet Protocol) fragmentados. En este caso, el atacante modifica el offset de dichos paquetes fragmentados, de manera que se solapen unos con otros, por tanto, el sistema operativo de la víctima no es capaz de reordenar los fragmentos, lo cual puede derivar en el colapso de este. [25]

3.1.2 Ataques DDoS

Los ataques distribuidos de denegación de servicio o ataques DDoS se diferencian principalmente de los ya expuestos ataques DoS, en el hecho de que, en este caso, son varias máquinas conjuntas, las que llevan a cabo el ataque. Principalmente se busca más que atacar una máquina en concreto, conseguir la denegación de servicio en recursos o servidores web, por ello requieren de conexión a internet de todas las máquinas implicadas. [7] [27] [29]

Para llevar a cabo estos ataques se han de seguir una serie de pasos. En primer lugar, se ha de detectar una máquina que posea una vulnerabilidad, la cual posteriormente se tomará control sobre ella, convirtiéndola en la máquina principal del ataque. Una vez se tiene el control de dicha máquina principal, se ha de reunir un gran número de máquinas zombis (botnet), es decir, máquinas infectadas, las cuales se encuentran bajo en control de la máquina principal. Esto se logra principalmente mediante el uso de malware, o directamente traspasando los controles de autenticación de dichas máquinas. [29]

En cuanto a lo que se refiere a los diferentes tipos de ataques DDoS que podemos encontrarnos, estos son básicamente los mismos que para DoS, salvando la diferencia de que en este caso se llevan a cabo de manera distribuida y simultanea desde diferentes máquinas. Si bien, existen casos particulares los cuales son propios de ataques DDoS, y a continuación se exponen algunos de ellos:

- **Ataque LAND:** basado igualmente en el protocolo TCP y mecanismo para el establecimiento de conexión entre dos máquinas (threeway handshake). En este caso,

las peticiones SYN tendrán dirección de origen y destino la del propio servidor o víctima, de esta manera se pretende crear un bucle de paquetes SYN/ACK, el cual consiga colapsar dicho servidor consiguiendo de esta manera una denegación de servicio. [27]

- **Slowloris:** consiste en un ataque altamente dirigido, basado en activar un servidor web con la finalidad de que este consiga colapsar otro de ellos. Se considera altamente dirigido, ya que el ataque ha de realizarse sin que esto afecte al resto de servicios o puertos que también se encuentran en el mismo entorno que el servidor afectado. Esto se consigue manteniendo el mayor número de conexiones abiertas durante el mayor tiempo posible, lo cual se logra mediante el envío constante de conexiones parciales. Dichas conexiones se encuentran formadas principalmente por cabeceras HTTP, pero en ningún caso se completa dicha petición. El servidor por tanto mantiene abiertas todas estas conexiones falsas, lo cual deriva en un desbordamiento del pool de conexiones (número de conexiones que el servidor puede mantener abiertas), logrando así que los usuarios legítimos no puedan establecer conexión con el servidor. [7]
- **Amplificación NTP (Network Time Protocol):** en ellos el atacante explota los servidores NTP, los cuales son de acceso público, con el fin de colapsar un servidor mediante el envío de paquetes UDP. Se considera un ataque con carácter de amplificación ya que este posee una ratio petición-respuesta entre 1:20 y 1:200, lo que significa que fácilmente se puede lograr un ataque DDoS de carácter volumétrico. Para conseguir dichas conexiones NTP podemos usar diferentes herramientas como Metasploit o datos provenientes del proyecto Open NTP. [7]
- **Inundación HTTP:** consisten en hacer uso aparentemente legítimo de las peticiones HTTP GET y POST, con el fin de atacar un servidor web o aplicación. Este tipo de ataques, no requieren del uso de paquetes mal formados, suplantación de IP o técnicas de reflexión, y requiere de un menor ancho de banda para conseguir derribar al objetivo del ataque. El ataque es cuanto más efectivo cuando se fuerza a la víctima a usar el mayor número de recursos para responder cada una de las peticiones individuales. [7]

Para finalizar, hacer especial mención a los ataques *Zero-day* o *Día cero*, término que engloba a todos aquellos nuevos ataques o los cuales se desconocían, y para los cuales aún no se ha lanzado un parche que los mitigue. Este término se ha vuelto especialmente conocido entre la

comunidad de atacantes, en la cual el comercio de este tipo de ataques se ha vuelto una práctica habitual. [7]

3.2 Herramientas

A lo largo de esta sección se explicarán de manera breve y concisa las diferentes herramientas de las cuales se hará uso para el desarrollo del proyecto, de esta manera se pretende hacer más sencillo el seguimiento de lo que se refiere a la parte experimental, de tal forma que el lector sepa de antemano la función que realiza cada una de estas herramientas.

3.2.1 Mininet

Para la creación del entorno controlado, se decidió hacer uso de la herramienta Mininet. Esta consiste en una imagen de máquina virtual que nos da acceso a un sistema operativo Linux Ubuntu, el cual nos crea de manera sencilla un entorno virtual realista, con un kernel real y códigos de conmutadores y aplicaciones a nuestra libre disposición. La ventaja de ella reside en la facilidad de su uso, se puede interactuar de manera sencilla con el entorno creado, personalizarlo a nuestro gusto, así como, exportarlo a hardware real en caso de que esto fuera necesario. [30]

Como añadido, ya que es una herramienta preparada para el trabajo con redes ficticias, muchas herramientas de gran utilidad nos vienen ya dadas, como es el caso de Wireshark; así como, posibilidad de conectar vía ssh (Secure Shell) numerosos terminales, desde los cuales se pueden llevar a cabo diferentes operaciones simultaneas todas ellas sobre el mismo entorno. Otra de las ventajas que nos proporciona es la facilidad para restaurar el entorno, con un simple comando de consola podemos resetear nuestro espacio virtual y devolverlo a su estado original.

Todas estas características ya mencionadas hicieron que acabase decantándome por su uso, ya que creo que es la mejor opción para llevar a cabo el estudio.

3.2.2 Docker

En cuanto a lo que incluir en el entorno controlado los elementos necesarios para llevar a cabo el ataque se refiere, se optó por la opción de utilizar la herramienta conocida como Docker. Como su propio nombre indica, dicha herramienta se basa en el uso de contenedores, de esta manera se pueden ejecutar diferentes aplicaciones simultáneamente en contenedores aislados, de tal forma que se obtiene una mejor capacidad de computación en la máquina. [31]

Para comprender como funciona Docker, en primer lugar, hemos de clarificar el concepto de contenedor. Un contenedor es principalmente un paquete ejecutable que contiene una imagen de pequeño peso, aislada y con el software necesario para poder trabajar con ella, ya sean herramientas, librerías, ajustes, etc.…Docker se encuentra disponible tanto para Linux como para Windows, siendo independiente el comportamiento del contenedor del sistema operativo sobre el que se ejecuta Docker. Así como, el uso de contenedores evita los posibles conflictos resultantes de ejecutar diferentes softwares en el mismo entorno. [32]

El uso de estos contenedores se enfocará principalmente a la creación de las maquinas atacantes y de las víctimas, además de para ejecutar diferentes servidores de DNS.

3.2.3 Hping3

Para poder llevar a cabo el ataque, se requiere de una herramienta que nos permita enmascarar la IP de origen del atacante, y hacer que para el servidor DNS dicha IP sea la de nuestra víctima. Para ello, disponemos de hping3, la cual es una aplicación de terminal que nos permitirá modificar dicha dirección IP por aquella que nosotros queramos. [33]

Además, cuenta con otro gran número de funciones que nos serán de gran utilidad, entre las cuales encontramos la posibilidad de elegir tanto el protocolo de transporte, como el puerto al que irán dirigidos los mensajes; la posibilidad de introducir a modo de firma cualquier archivo de texto que se desee; y una de las herramientas de quizás mayor interés, la posibilidad de generar múltiples peticiones a una misma dirección IP desde diferentes IPs de origen aleatorias de manera tal que se envían paquetes a tiempo real de forma masiva consiguiendo simular un ataque DDoS. [33] [34]

3.2.4 Wireshark

Herramienta analizadora de protocolos, la cual nos permitirá observar a tiempo real y con gran profundidad todo aquello que suceda en nuestro entorno de tal manera que se podrá analizar el tráfico de paquetes y comprobar así de manera clara y sencilla las IPs de origen y destino de estos en los diferentes interfaces de nuestra red. [35]

Como característica de gran valor, se ha de destacar la calidad de la interfaz gráfica de la aplicación, motivo por el cual nos decantamos por el uso de esta respecto a otras disponibles.

3.2.5 Htop

Htop es una aplicación a tiempo real para sistemas operativos Linux, la cual nos permite monitorizar aquellos procesos que estén teniendo lugar en nuestra máquina. Gracias a ella, podemos controlar los niveles de consumo de CPU y memoria, lo cual nos permitirá saber si el ataque está siendo efectivo o no en función de dichos valores. [36] [37]

Se decidió optar por htop frente a otras alternativas como top, ya que esta es mucho más visual y amigable para el usuario. El uso de colores, el indicar los niveles de consumo en valores porcentuales y la distribución en pantalla de los procesos hace que su uso sea mucho más sencillo frente a las otras aplicaciones. Si bien, esta nos da la información de manera más reducida y no tan completa, para el uso que se requiere no es de importancia.

3.2.6 Docker Stats

Como complemento a htop para aquellos casos más complejos en los cuales necesitamos determinar de manera más precisa el porcentaje de CPU y memoria RAM que se encuentra empleando cada contenedor, haremos uso de Docker Stats. [38]

Ello nos permitirá observar el desglose total de estas variables y analizar cuál es el contenedor que se encuentra consumiendo la mayor parte de los recursos, de esta manera junto con htop podemos llevar a cabo un análisis más preciso y correcto del desarrollo de los ataques.

3.2.7 Edraw Max

Para la realización de los diagramas que mostrarán los esquemáticos de las redes en las cuales se va a trabajar, se empleará el software Edraw Max, el cual nos permite crear esquemáticos de carácter profesional y de manera sencilla. La opción *Network Diagram* nos aporta todos los elementos necesarios para la realización de dichos esquemáticos, además de diferentes opciones de diseño para cada elemento, ajustándose por tanto a cualquier necesidad que pueda tener el usuario. Pese a que es un software de pago, se hará uso de la versión gratuita ya que con ella nos es suficiente para las necesidades que se requieren. [39]

Capítulo 4

4 Diseño

Como todo proyecto de experimentación, se requiere del diseño de una arquitectura a partir de la cual se pueda establecer un prototipo para llevar a cabo las pruebas deseadas. Tratándose de llevar a cabo la simulación controlada de un ataque de denegación de servicio basado en servidores DNS, se requiere de la creación de un gran entorno, en el cual se han de incluir numerosos elementos. Dichos elementos incluyen terminales de usuario, routers, servidores DNS en la red, etc…

A continuación, se muestra una imagen la cual representa un esquemático de lo que sería el diseño ideal para llevar a cabo la experimentación y estudio de dichos ataques.

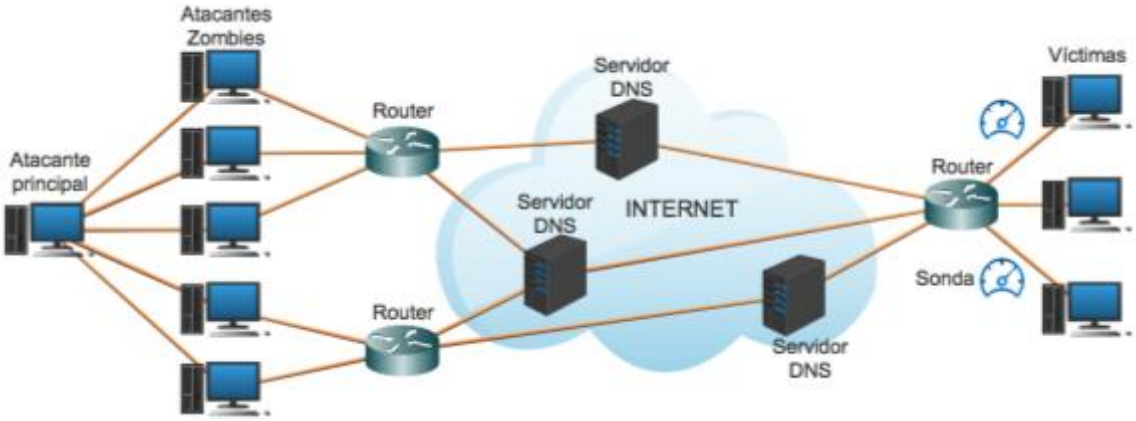


Figura 7: Diseño

- En primer lugar, se requiere de una máquina que haga las funciones de atacante principal, esta será la encargada de explotar las vulnerabilidades de otras máquinas, consiguiendo reunir un gran conjunto de estas, las cuales actuarán a su vez de atacantes zombis.
- En cuanto al conjunto de máquinas o atacantes zombis, hemos de reunir un número de equipos muy superior al de víctimas, pongamos como número genérico M. El número genérico que representará el número de víctimas será N.
- A su vez, se requiere de una serie de routers, los cuales permitan conectar los diferentes equipos a los servidores DNS que se encuentran en la red. Llegados a este punto, conviene puntualizar que, para llevar a cabo una simulación más realista, las máquinas que actúan de atacantes zombis se agruparan en bloques, de tal forma que se dividan en diferentes zonas o dominios DNS.

- En cuanto a los servidores DNS, estos estarían situados en la red, y se haría uso de varios de ellos, de tal forma que se consiga un gran volumen de peticiones y se reparta el tráfico entrante entre ellos. Además, como en el punto anterior se ha indicado, se haría uso de diferentes zonas o dominios, las cuales tendrían sus propios servidores DNS asignados.
- Finalmente, en los equipos de las víctimas, se colocarían sondas, cuya finalidad consiste en medir el tráfico resultante de la simulación del ataque. De esta manera se podría observar y comparar con el tráfico obtenido previo a la simulación y, por tanto, intentar llegar a tasar y fijar el volumen de tráfico requerido para conseguir llevar a cabo la denegación de servicio en dichas máquinas.

Hacer de nuevo hincapié en la relación que debe existir entre máquinas atacantes frente a máquinas víctimas. Previamente se han fijado dos valores genéricos M y N correspondientes a atacantes y víctimas respectivamente, y como se ha comentado M ha de ser mucho mayor que N (M>>N).

Como resultado del estudio bajo este diseño, se pretende establecer y fijar bajo que volumen de tráfico y condiciones se consigue denegar servicio a una máquina de determinadas características, así como, una vez conseguida la denegación de servicio, tratar de poner en práctica métodos de prevención, con el fin de encontrar una solución frente a este tipo de amenazas.

Capítulo 5

5 Prototipo

En el apartado anterior se ha llevado a cabo la presentación de lo que sería el diseño de un entorno idílico para el desarrollo del estudio, si bien, para el desarrollo de este se requería de un presupuesto altamente elevado y de una infraestructura fuera de nuestro alcance. Por ello a lo largo de esta sección, se presentarán dos opciones de prototipos bajo los cuales se podría llevar a cabo el estudio de una manera mucho más sencilla y sin problemas en cuanto a términos legales y a presupuesto se refiere.

5.1 Prototipo hardware real

En primera instancia se pensó en desarrollar un prototipo basado en el uso de hardware físico, fundamentado principalmente en una simplificación del diseño idílico ideado. De tal manera que se pudieran seguir logrando los objetivos marcados al inicio del proyecto sin que esto afectase a ellos.

En primer lugar, llevar a cabo la simulación en lo que al espacio de internet se refiere supondría tener la propiedad de servidores DNS, sumado a diferentes dominios en los cuales alojar las diferentes máquinas, ya que como se ha expuesto, estas se situarían en dominios y zonas DNS diferentes. Para evitar estos inconvenientes, el estudio podría desarrollarse a nivel local, en una red interna controlada, creada específicamente para llevar a cabo el estudio. De esta manera también evitamos el riesgo que supone lanzar un ataque de denegación de servicio a través de la red.

Otro de los posibles elementos a suprimir, sería la existencia del atacante principal, ya que nuestro objetivo se centra en observar el desarrollo del ataque en la víctima, y por tanto incluir dicho elemento complica en sobremanera el estudio.

De esta manera, se obtiene ya un prototipo mucho más sencillo de implementar, ya que la construcción de dicho entorno a nivel local es relativamente sencilla. Tratándose de un desarrollo en LAN (Local Area Network), los diferentes elementos que componen nuestra red podrían conectarse de manera física, de tal manera se reduciría también así la dificultad conjunta de ello. A continuación, se muestra una imagen en la cual se representa el esquemático de lo que sería dicho prototipo en hardware real.

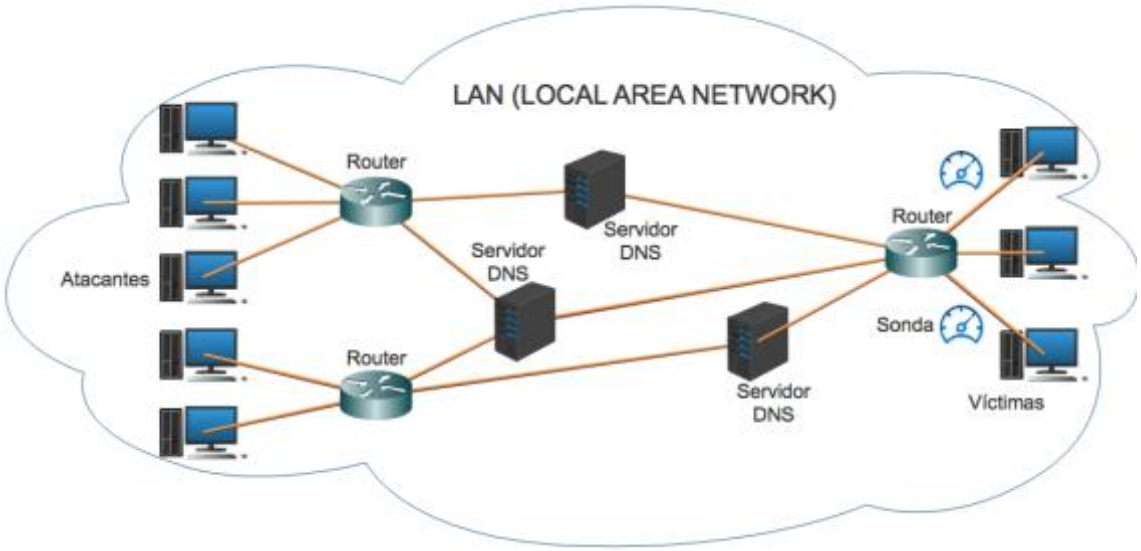


Figura 8: Prototipo hardware real

5.2 Prototipo virtual

Ya que la posibilidad de llevar a cabo un prototipo real no era posible, se decidió llevar a cabo un prototipo virtual, sobre el cual se realizará el estudio. Para el montaje de dicho prototipo, se emplearán algunas de las herramientas previamente mencionadas en su correspondiente sección. Dichas herramientas necesarias serán Mininet y Docker, y su proceso de instalación será detallado a continuación.

5.2.1 Instalación Mininet

Como ya se explicó previamente, Mininet es una imagen que ha de ser montada sobre un soporte que nos permita su uso. Para ello emplearemos Fusion, plataforma de virtualización de sistemas operativos proporcionada por VMWare y exclusiva para Mac. Para la obtención de la imagen, fue tan sencillo como la descarga directa desde la página oficial de la organización (http://mininet.org/download/), en ella además se nos proporciona una guía con pasos a seguir en su instalación.

En cuanto a los ajustes asignados a la máquina virtual, se decidió proporcionarle dos de los cuatro núcleos de los cuales dispone el ordenador, así como más de la mitad de la memoria RAM (Random Access Memory), 5GB. Además, se dotó a esta de dos adaptadores de red, el primero de ellos conectado a la máquina original (host); el segundo de ellos a la red de Internet empleando NAT (Network Address Translation), el cual permite a redes IP privadas que emplean direcciones no registradas conectarse a Internet, por tanto, dicho adaptador de red funciona de

tal manera que traduce la dirección IP privada de nuestro entorno y la cual no es única, en una dirección legal. [40]

De esta manera, ya tenemos listo el entorno Mininet para proceder a la creación de nuestra red sobre la cual llevaremos a cabo el ataque.

5.2.2 Instalación Docker. Contenedores

La obtención de Docker para nuestro sistema es relativamente sencilla, lo único a tener en cuenta sería la versión a obtener, ya que existe una versión gratuita para toda la comunidad y otra versión de pago para empresas.

En la página principal, se ha de seleccionar la obtención de Docker para servidores Ubuntu, posteriormente seleccionar la opción CE (Community Edition) y tras ello, clicar en el enlace que nos aparece, todo este proceso nos redirige al tutorial de instalación que hemos de seguir (https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/).

Una vez Docker ya se encuentra corriendo en nuestro sistema, se procede a la inclusión de los contenedores en este. Como se propuso anteriormente en el texto, se necesitarán contenedores tanto para el servidor/es DNS como para las máquinas de los atacantes y de la víctima.

Cabe destacar el hecho de que, al tratarse de una virtualización, el uso de los recursos destinados al conjunto del entorno es compartido, es decir, tanto la CPU como la memoria RAM son empleadas a la vez por los diferentes elementos de los que se dispone. Por ello, a la hora del montaje de los diferentes contenedores, se les asignará un máximo de CPU a utilizar, de esta manera se trata de conseguir una distribución más equitativa de dichos recursos, ya que en un caso real los atacantes requieren de un porcentaje de cómputo mucho menor que la víctima. Así como, los servidores DNS son equipos los cuales con un porcentaje de cómputo relativamente bajo son capaces de lidiar con grandes cantidades de tráfico, ya que se trata principalmente del procesado de paquetes UDP.

Para ello en una primera aproximación se ha decidido dividir dicha capacidad de cómputo en un 25% para el servidor DNS, un 35% para la víctima y un 10% para los atacantes. Como se puede observar, la suma de los porcentajes del servidor DNS y de la víctima asciende al 60%, el 40% restante se empleará en aumentar en número de atacantes en las sucesivas pruebas que se irán realizando.

5.2.2.1 Montaje servidor DNS

En lo que al servidor DNS se refiere, en primera instancia se decidió optar por un servidor de tipo BIND (Berkeley Internet Domain Name), ya que este es el más común principalmente entre los sistemas Unix y en lo que Internet se refiere. De entre todos los contenedores disponibles se decidió escoger el proporcionado por el usuario sameersbn (https://github.com/sameersbn) [41], ya que en principio se ejecutaba correctamente y además nos proporciona la posibilidad de acceder al contenedor ejecutando el comando bash, de esta manera podíamos modificar aquellos parámetros que creyésemos necesarios, así como, observar el comportamiento interno del servidor.

Si bien, cuando se llevaron a cabo pruebas más exhaustivas se observó que el comportamiento no era del todo correcto, el servidor pese a estar activo no respondía adecuadamente al comando *dig*, y, por tanto, no era capaz de devolver los servidores de nombre asociados a este. Esto derivaba en la imposibilidad de obtener los bytes asociados a dicha petición de tipo DNS, los cuales nos serán necesarios para el desarrollo del proyecto. Ante esta situación se decidió buscar otro servidor DNS que cumpliese los requisitos necesarios.

Tras llevar a cabo varias búsquedas, finalmente se encontró un contenedor que nos proporcionaba un servidor DNS, el cual aun no siendo de tipo BIND, funciona correctamente y nos permite desarrollar el trabajo sin ningún tipo de inconvenientes. Dicho contenedor se encuentra conformado tanto por un servidor Dnsmasq como por un generador de ficheros a partir de plantillas el cual emplea los metadatos del contenedor. Si bien, para nuestros intereses únicamente hemos de centrarnos en lo que al servidor Dnsmasq se refiere, este tipo de servidores proporcionan al mismo tiempo servicios tanto de servidor DNS como de servidor DHCP (Dynamic Host Configuration Protocol), siendo estos principalmente diseñados para dar servicios a redes de pequeña dimensión (menos de cincuenta ordenadores) [42]. Por todo ello, sumado a su correcto funcionamiento frente a nuestras necesidades, se decidió optar por esta opción. Cabe mencionar al desarrollador de dicho contenedor, el usuario jderusse (https://github.com/jderusse/). [43]

A continuación, se muestra el comando ejecutado para el montaje de dicho servidor DNS, además, se procede a la explicación de los parámetros que lo componen.

docker run --detach --name dns -c 256 --publish 172.17.0.1:53:53/udp --volume /var/run/docker.sock:/var/run/docker.sock jderusse/dns-gen

- El parámetro *–detach* permite ejecutar en segundo plano el contenedor y nos devuelve su ID.
- *--name* nos sirve para nombrar como deseemos al contenedor, dándole el nombre que nosotros queramos.
- *-c* es el parámetro que nos permite seleccionar que cantidad máxima de CPU puede usar dicho contenedor, en este caso 256, ya que se ha de llevar a cabo respecto al valor de referencia 1024, por tanto, el 25% de 1024 se traduce en 256. [44] [45]
- *--publish 172.17.0.1:53:53/udp* hace que el puerto 53:53 del protocolo udp en la dirección IP 172.17.0.1 sea visible por el host, de esta manera, ya que se trata de un servidor dnsmasq, conseguiremos conexión entre nuestro servidor y los servidores de nombre que se ejecutan en segundo plano; permitiendo así el buen funcionamiento de nuestro servidor.
- *--volume* indica el lugar en el cual se encuentra almacenado el volumen que contiene la imagen del contenedor a montar.

Posteriormente, hemos de incluir la dirección 172.17.0.1 dentro del fichero que contiene los diferentes servidores de nombre a los cuales nuestro servidor DNS acudirá cuando reciba una petición, para ello hemos de modificar el fichero *resolv.conf.d* [43]

echo "nameserver 172.17.0.1" | sudo tee --append /etc/resolvconf/resolv.conf.d/head

Una vez ejecutado dicho comando ya hemos modificado el fichero y únicamente hemos de reiniciarlo para que sea visible la modificación. [43]

sudo resolvconf –u

Tras llevar a cabo todos estos pasos, ya se puede decir que se ha montado correctamente el servidor DNS y que este se encuentra listo para su uso.

Llegados a este punto, se empleó dicho servidor de tipo dnsmasq para la realización de las primeras pruebas del estudio, si bien, en un momento dado, se observó un comportamiento anómalo en su funcionamiento, por ello, se decidió volver al servidor de tipo BIND previamente empleado. Como se ha comentado anteriormente, dicho servidor no operaba correctamente debido a un error de configuración que en su momento no se supo corregir, pero dadas las circunstancias en las que nos encontrábamos, se decidió tratar de configurarlo correctamente con el fin de hacer uso de este.

Finalmente se consiguió configurar y que este respondiese de manera correcta a las peticiones DNS que se le realizaban desde el atacante. Por ello para la última prueba del estudio se empleará este nuevo servidor.

5.2.2.2 Montaje equipos

En cuanto a los equipos que actuarán tanto de atacantes como de víctima, se emplearán contendores que nos proporcionarán un sistema operativo Linux Ubuntu. Uno de los requisitos principales para la elección del contenedor era que en este se debía de poder ejecutar el comando bash, pudiendo así acceder al interior del contenedor, siendo esto necesario tanto para el lanzamiento del ataque como para la monitorización de los recursos empleados por víctima. El contenedor elegido finalmente fue el proporcionado por el usuario sameersbn [46], pese a que su servidor DNS no nos fue finalmente útil en primera instancia, en este caso los contenedores que nos proporcionaban la distribución de Ubuntu funcionaron sin problemas en las diferentes pruebas previas que se realizaron. Entre estas pruebas se encontraban, por ejemplo, el lanzamiento de pings entre los diferentes componentes del entorno para comprobar si existía comunicación entre ellos, la posibilidad de instalar nuevos recursos en las máquinas (se requería la instalación tanto de hping3, como de htop), etc…

A continuación, al igual que se ha hecho con el servidor DNS, se va a proceder a mostrar los comandos ejecutados para llevar a cabo el montaje de los equipos atacantes y de la víctima. En primer lugar, se mostrará el comando referente a la víctima.

docker run -itd --name victima -c 359 sameersbn/ubuntu

- *-itd* es un comando conjunto que aúna en si tres comandos independientes. En primer lugar *–i* nos permite mantener una conexión interactiva, es decir, la conexión STDIN (Stardard Input) se mantiene abierta independientemente de si nos encontramos adjuntos al contenedor. La opción *–t* nos permite visualizar los logs del contenedor en nuestro terminal [47]. Y finalmente *–d* es la versión abreviada del comando *–detach* que se ha explicado anteriormente.
- De nuevo observamos el parámetro *–c* en este caso con valor 359. Como se ha comentado previamente, al contenedor de la víctima se le asignaría un valor relativo al 35% de la capacidad total de cómputo de la CPU (1024 es el valor de referencia), lo que se traduce en 358.4. Ya que el parámetro únicamente acepta valores enteros, se decidió optar por 359 en vez de por 358. [43] [44]
- *sameersbn/ubuntu* se corresponde con el volumen a montar.

Ahora procedemos a observar el comando empleado para el montaje del atacante, este es idéntico al empleado para la víctima, salvo por las diferencias en cuanto al nombre asignado al contenedor, y el valor de CPU asignado, que en este caso se corresponde con un 10% de 1024, que, tras su aproximación para ser un valor entero, se fija en 103.

docker run -itd --name atacante -c 103 sameersbn/ubuntu

Como resultado se obtendría un entorno como el que se muestra en la siguiente imagen, el cual muestra una única maquina atacante y un solo servidor de DNS.

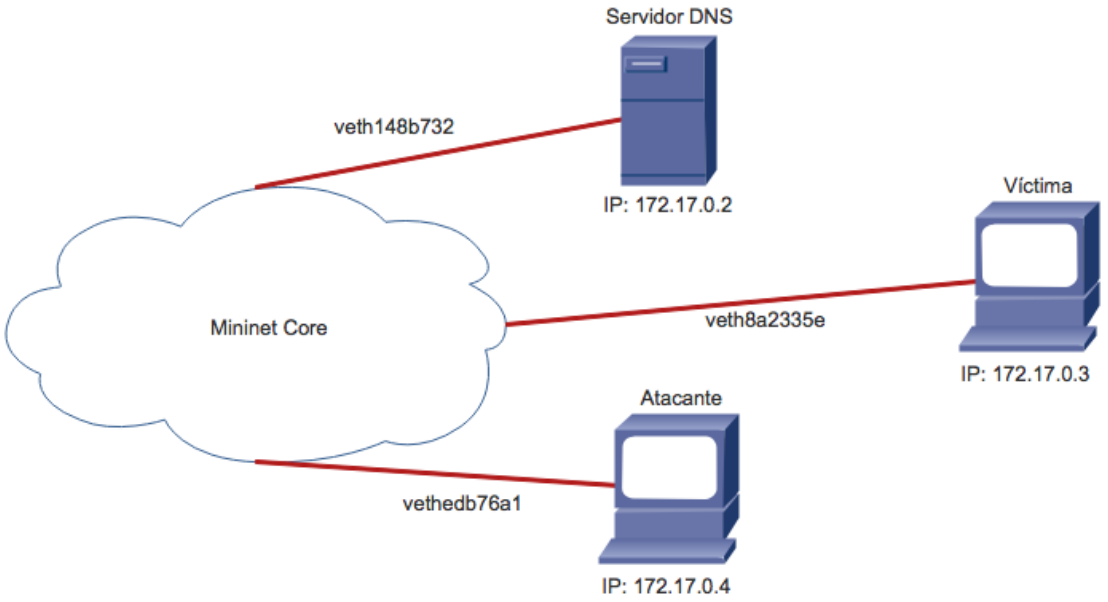


Figura 9: Ejemplo entorno

En la siguiente imagen se muestra el resultado de ejecutar el comando *ifconfig* en el terminal principal de Mininet, bajo la misma configuración anteriormente mostrada, de esta manera, podemos observar todas las interfaces que se encuentran conectadas a ella.

```
root@mininet-vm:/home/mininet# ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:58:40:b7:0a
          inet addr:172.17.0.1  Bcast:0.0.0.0  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0      Link encap:Ethernet  HWaddr 00:0c:29:5d:82:f7
          inet addr:192.168.248.129  Bcast:192.168.248.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1087 errors:0 dropped:0 overruns:0 frame:0
          TX packets:846 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:99708 (99.7 KB)  TX bytes:101553 (101.5 KB)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:5d:82:01
          inet addr:172.16.129.146  Bcast:172.16.129.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5552 (5.5 KB)  TX bytes:5741 (5.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:17730 (17.7 KB)  TX bytes:17730 (17.7 KB)

veth148b732 Link encap:Ethernet  HWaddr 9e:1a:88:29:c9:2c
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth8a2335e Link encap:Ethernet  HWaddr e2:b4:8d:8c:0c:19
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vethedb76a1 Link encap:Ethernet  HWaddr 1a:10:9d:92:d1:ef
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figura 10: Interfaces ejemplo

Capítulo 6

6 Pruebas. Resultados

Una vez finalizado el proceso de montaje del prototipo, se procede a la realización de las pruebas y, por tanto, de tratar de simular un ataque de denegación de servicio. Con el fin de observar como varían los resultados en función del número de máquinas atacantes, así como, del número de servidores DNS empleados, se dividieron las pruebas en varias fases.

6.1 1 Atacante / 1 Servidor DNSMASQ

En esta fase se llevará a cabo la simulación de un ataque de denegación de servicio DoS que emplea una única máquina atacante y un solo servidor de DNS.

En primer lugar, observamos el rendimiento que tiene la victima previamente al ataque, de esta manera podremos comparar de manera verídica si el ataque funciona o no realmente. Para ello haremos uso del comando *htop*, el cual nos muestra entre otras cosas, el porcentaje de computo que tiene la máquina en tiempo real, y como se encuentra este repartido entre usuario y sistema, además de los diferentes procesos que se están llevando a cabo.

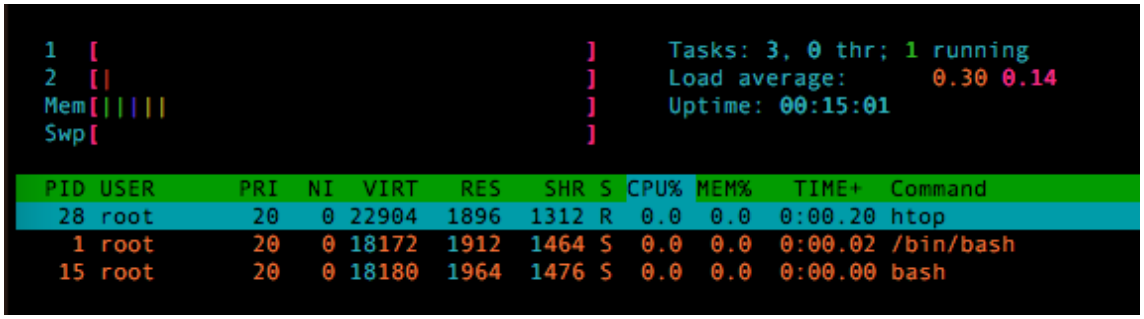


Figura 11: Rendimiento previo víctima primera prueba

Como se puede observar, la máquina posee unos porcentajes de computo en CPU (determinado por los valores 1 y 2 de la parte superior izquierda) y de uso de memoria RAM muy bajos, ya que no se encuentra realizando ninguna tarea. Esto nos indica que la máquina se encuentra en valores coherentes y por tanto podemos proseguir con la prueba.

Como apunte, el último valor indicado se corresponde con la memoria Swap, esta es, aquella memoria disponible en el disco duro en caso de que la máquina necesite almacenar una cantidad de información superior a la que la memoria RAM puede hacer frente.

Como previamente se ha comentado, nuestro entorno se encuentra construido bajo una virtualización, esto significa que todos los recursos son compartidos entre los diferentes elementos que lo componen. Por ello a la hora de llevar a cabo la simulación del ataque, hemos de minimizar el número de procesos que se encuentren realizándose fuera de lo que son las máquinas involucradas, ya que esto afectaría a nuestros valores, alejándolos de la realidad. Debido a esto, se han de realizar dos simulaciones para cada fase, una en la cual se observe el ataque en sí, y si este es efectivo o no; y otra en la cual se llevará a cabo la captura de paquetes por medio de Wireshark.

Previo al ataque, necesitamos exportar en un fichero la consulta DNS que queremos enviar. Dicha consulta nos es de gran utilidad ya que la respuesta recibida por la víctima tendrá un tamaño alrededor de cuatro veces mayor respecto a la petición llevada a cabo por el atacante, lo cual aporta un carácter de amplificación al ataque haciendo que este sea mucho más efectivo. [48]

Source	Destination	Protocol	Length	Info
172.17.0.4	172.17.0.2	DNS	70	Standard query 0xc139
172.17.0.2	172.17.0.4	DNS	281	Standard query response

Figura 12: Consulta DNS primera prueba

Como se aprecia en la imagen, el tamaño de la petición es de 70 bytes mientras que la respuesta posee una longitud de 281, quedando así demostrado el efecto de amplificación que se busca al exportar dicha petición en un fichero.

6.1.1 Desarrollo del ataque

Una vez realizados estos pasos previos llega el momento de llevar a cabo el ataque, para ello desde la consola del atacante se lanzará un comando hping3 seguido de una serie de parámetros que hemos de establecer según nuestras necesidades. El comando en sí se muestra a continuación y se procede a su explicación [34]:

hping3 --faster --udp -p 53 --spoof 172.17.0.3 --file query -d 28 172.17.0.2

- El parámetro *–faster* hace que los paquetes se envíen de manera continuada en intervalos de aproximadamente un microsegundo, o lo que es lo mismo, un millón de paquetes por segundo.
- Los parámetros *–udp –p 53* indican el protocolo mediante el cual se envían los paquetes, y el puerto de destino al que se envían.

- El parámetro *–spoof 172.17.0.3* sirve para llevar a cabo la suplantación de IP, colocando como IP de la víctima como la originaria de los paquetes.
- Los parámetros *–file query –d 28* nos permiten enviar el contenido del fichero seleccionado como contenido de la consulta a realizar. El contenido de dicho fichero es la consulta DNS previamente exportada.
- Finalmente, el parámetro *172.17.0.2* indica la dirección IP del servidor DNS al cual se quiere enviar la consulta.

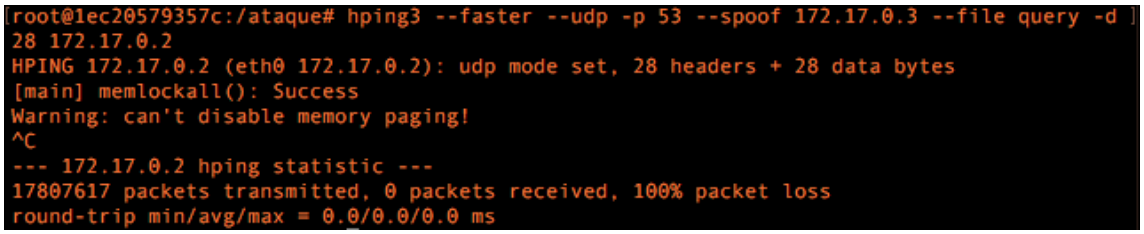


Figura 13: Lanzamiento prueba hping3

En la imagen podemos apreciar el lanzamiento del comando hping3 y el número de paquetes transmitidos, los cuales son 17.807.617. Como cabe de esperar la estadística nos devuelve un 100% de paquetes perdidos, eso es debido a que la dirección IP de origen del paquete ha sido modificada por la de la víctima gracias al parámetro *–spoof.* A continuación, analizaremos si el ataque ha sido efectivo y ha discurrido de la manera adecuada.

En primer lugar, comprobamos el estado de la memoria RAM y de computo de CPU en los instantes iniciales, de esta manera observamos si la victima comienza a procesar la llegada de los paquetes. También es de importancia el valor *Uptime*, ya que este nos servirá para determinar cuánto tiempo ha discurrido desde el lanzamiento hasta el momento en el cual se consigue saturar a la víctima haciendo efectivo el ataque.

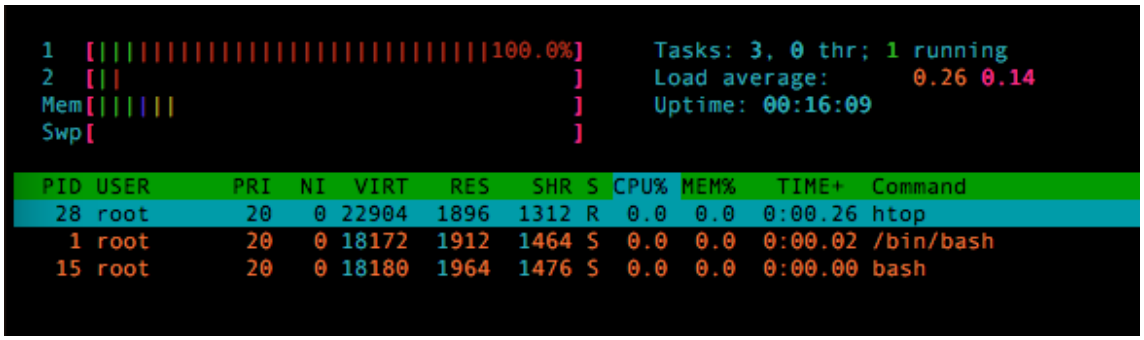


Figura 14: Estado inicial víctima primera prueba

Como se puede observar, la víctima ha comenzado a procesar los paquetes, ya que uno de los dos núcleos de CPU de los que dispone se encuentra trabajando al máximo de sus posibilidades.

En el caso de la memoria RAM, esta se encuentra en valores bajos, por lo que la máquina tiene margen aún antes de denegarle el servicio al usuario. En cuanto al valor *Uptime*, es de 00:16:09.

Transcurridos seis minutos se decide detener el ataque y observar los diferentes valores en dicho instante de tiempo.

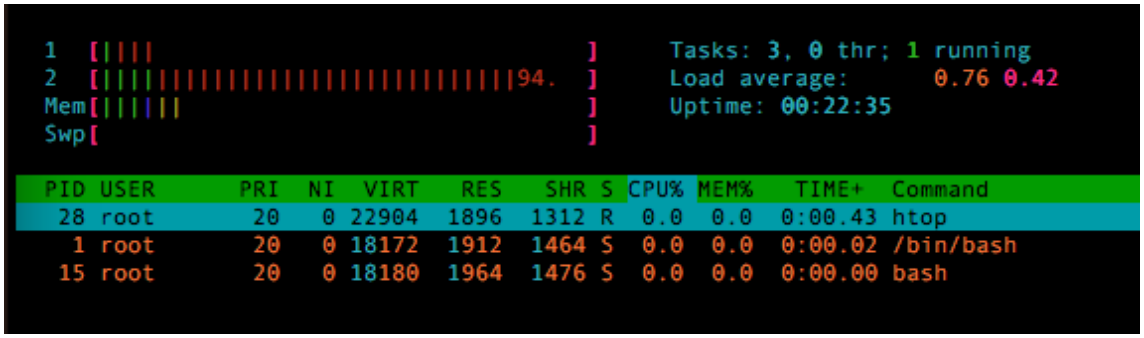


Figura 15: Estado final víctima primera prueba

En este caso, se aprecia como el nivel de cómputo de los núcleos de CPU ha variado en su distribución, ahora es el núcleo dos el cual se encuentra operando a alto rendimiento mientras que el uno se encuentra en niveles bajos. Independientemente de esto, el nivel global de computo es similar, el nivel de recursos empleados es prácticamente el mismo, y por tanto el efecto sobre la víctima será el mismo. En lo relativo a la memoria RAM, esta ha permanecido inmóvil, valor que era de prever ya que el nivel de computo de la CPU en su conjunto no se encontraba nunca a su nivel máximo, ni en valores cercanos a este.

En el caso de la memoria Swap, el valor es nulo, esto es obvio ya que la memoria RAM no se encuentra saturada, y por tanto no existe la necesidad de derivar esa necesidad computacional al disco duro.

Obtenidos estos valores en este lapso de tiempo, podemos decir que el ataque no ha sido efectivo. En ningún momento se ha conseguido un nivel de computo de CPU cercano al máximo de sus posibilidades, siendo de esta manera imposible conseguir llevar a cabo el ataque. La máquina víctima del ataque era capaz de procesar todos los paquetes sin aumentar notoriamente su actividad.

6.1.2 Análisis del tráfico

A continuación, se muestran diferentes capturas de Wireshark, en las cuales se muestra el tráfico de paquetes en las diferentes interfaces.

6.1.2.1 Tráfico interfaz atacante

En primera instancia analizaremos el tráfico en la interfaz del equipo atacante, con el fin de conocer si el ataque se ejecuta correctamente desde su punto inicial.

Source	Destination	Protocol	Length	Info
172.17.0.3	172.17.0.2	DNS	70	Standard query
172.17.0.3	172.17.0.2	DNS	70	Standard query
172.17.0.3	172.17.0.2	DNS	70	Standard query
172.17.0.3	172.17.0.2	DNS	70	Standard query

Figura 16: Tráfico atacante primera prueba

En la imagen se aprecia como en la interfaz correspondiente al atacante, se observa tráfico de salida cuya IP de origen es la de la víctima, y la IP de destino es el servidor DNS. Además, se observa que dicho tráfico corresponde a una petición DNS, y que la longitud en bytes de esta es de 70. Por tanto, en esta interfaz se observa un comportamiento correcto.

6.1.2.2 Tráfico interfaz servidor DNS

La complejidad reside principalmente en analizar el tráfico en el servidor DNS, ya que es donde confluye el tráfico entrante del atacante, con el tráfico saliente hacia la víctima, lo que hace que su estudio sea más detallado y laborioso. Para ello se hará uso de las expresiones de filtrado que Wireshark nos proporciona, de esta manera podremos visualizar de manera más sencilla el tráfico que nos interesa analizar.

Primeramente, observaremos el tráfico con IP de origen la de la víctima e IP de destino la del servidor DNS. En este caso se empleará la expresión *ip.src==172.17.0.3 and ip.dst==172.17.0.2* para filtrar los paquetes.

Source	Destination	Protocol	Length	Info
172.17.0.3	172.17.0.2	DNS	70	Standard query 0xf4ab
172.17.0.3	172.17.0.2	ICMP	309	Destination unreachable
172.17.0.3	172.17.0.2	DNS	70	Standard query 0xf4ab

Figura 17: Tráfico DNS 1 primera prueba

Analizando la imagen podemos observar cómo se han capturado dos tipos de paquetes que poseen como dirección IP de origen la de la víctima. El primero de ellos, de protocolo DNS, corresponde con los paquetes con la IP modificada lanzados desde el atacante los cuales se encuentran llegando al servidor, esto es posible saberlo gracias a la longitud de estos, ya que siendo de 70 bytes nos indican que es una petición DNS y no una respuesta. El segundo tipo de

paquete que se observa, de tipo ICMP, es consecuencia de dicha comunicación entre servidor y víctima, la cual esta no se espera y por tanto envía dichos paquetes con el fin de conocer el por qué se encuentra recibiendo tal cantidad de respuestas DNS.

Momento ahora de analizar el tráfico saliente del servidor DNS y con destino la víctima, para ello se aplica el filtro *ip.src==172.17.0.2 and ip.dst==172.17.0.3*

Source	Destination	Protocol	Length	Info
172.17.0.2	172.17.0.3	DNS	281	Standard query response
172.17.0.2	172.17.0.3	DNS	281	Standard query response
172.17.0.2	172.17.0.3	DNS	281	Standard query response
172.17.0.2	172.17.0.3	DNS	281	Standard query response

Figura 18: Tráfico DNS 2 primera prueba

En la imagen se aprecia como dicho tráfico se corresponde con las respuestas a las peticiones DNS llegadas, esto es posible saberlo, no sólo por la información que se nos proporciona en su correspondiente campo de la tabla, sino también por el tamaño de dichos paquetes, siendo estos de 281 bytes. Ya que la IP de origen de las peticiones había sido modificada, el servidor responde no al legítimo creador de las peticiones, sino a la IP que aparecía como originaria de estas.

En mitad de todo el proceso previamente analizado, ocurre un paso intermedio, en el cual nuestro servidor de DNS mantiene una comunicación bidireccional con los verdaderos servidores de nombre (nameservers), que serán ellos quienes procesen en primera instancia tanto la petición como la respuesta. Para ello debemos conocer cuáles son las direcciones a las que nuestro servidor DNS se va a referir, esto se consigue acudiendo al directorio */etc* del propio entorno de Mininet y observando el fichero *resolv.conf.* [43]

Este comportamiento se debe a la utilización de un servidor DNS de tipo dnsmasq, el cual ha de recurrir a los servidores de nombre que se encuentran por detrás de él para así procesar las peticiones que le llegan. Esto solo ocurre la primera vez que recibe una petición, ya que, tras consultar a los ya mencionados servidores de nombre, nuestro servidor dnsmasq almacena en su memoria caché la respuesta obtenida por parte de estos y para peticiones posteriores no es necesaria dicha comunicación, únicamente comprueba su memoria caché y emite la respuesta directamente al equipo destinatario de la petición.

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.248.1
nameserver 172.16.129.2
nameserver 172.17.0.1
search localdomain
~
~
```

Figura 19: Dirección servidores de nombre

Como se ve en el fichero, las direcciones de los servidores de nombre a los cuales recurrirá nuestro servidor DNS son 192.168.248.1, 172.16.129.2 y 172.17.0.1. Con el fin de agilizar el proceso, solo analizaremos el caso 172.17.0.1

Source	Destination	Protocol	Length	Info
172.17.0.2	172.17.0.1	DNS	70	Standard query 0x0ae6 NS <Root>
172.17.0.1	172.17.0.2	DNS	70	Standard query 0x0ae6 NS <Root>
172.17.0.2	172.17.0.1	DNS	281	Standard query response 0x3332 NS a.root-servers.net
172.17.0.1	172.17.0.2	DNS	281	Standard query response 0x3332 NS a.root-servers.net

Figura 20: Tráfico DNS 3 primera prueba

La comunicación bidireccional se aprecia fácilmente, ya que ambos mensajes se encuentran de manera consecutiva, y analizando el campo información vemos que se trata de la misma consulta. En cuanto a las respuestas, podemos observar los diferentes servidores de nombre que han procesado y creado la respuesta a la petición DNS generada por el atacante.

Con el fin de conocer los servidores de nombre disponibles para procesar las peticiones, hemos de realizar una consulta DNS a nuestro servidor, el cual nos devolverá una respuesta con todos ellos. Dicha consulta realizada se muestra a continuación:

dig @172.17.0.2

```
root@1ec20579357c:/# dig @172.17.0.2

; <<>> DiG 9.9.5-3ubuntu0.15-Ubuntu <<>> @172.17.0.2
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29223
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 1280
;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       5       IN      NS      m.root-servers.net.
.                       5       IN      NS      b.root-servers.net.
.                       5       IN      NS      i.root-servers.net.
.                       5       IN      NS      k.root-servers.net.
.                       5       IN      NS      f.root-servers.net.
.                       5       IN      NS      c.root-servers.net.
.                       5       IN      NS      e.root-servers.net.
.                       5       IN      NS      d.root-servers.net.
.                       5       IN      NS      a.root-servers.net.
.                       5       IN      NS      h.root-servers.net.
.                       5       IN      NS      j.root-servers.net.
.                       5       IN      NS      g.root-servers.net.
.                       5       IN      NS      l.root-servers.net.

;; Query time: 64 msec
;; SERVER: 172.17.0.2#53(172.17.0.2)
;; WHEN: Thu Aug 31 23:32:34 UTC 2017
;; MSG SIZE  rcvd: 239
```

Figura 21: Consulta DNS 2 primera prueba

6.1.2.3 Tráfico interfaz víctima

Finalmente, pasamos a analizar el tráfico en la interfaz de la víctima, el cual no debe suponer ningún contratiempo, ya que este ha de ser idéntico al saliente del servidor DNS con destino dicha máquina. Además, se han de observar paquetes ICMP salientes, como resultado de dicha comunicación no esperada entre víctima y servidor.

Source	Destination	Protocol	Length	Info
172.17.0.2	172.17.0.3	DNS	281	Standard query response
172.17.0.3	172.17.0.2	ICMP	309	Destination unreachable
172.17.0.2	172.17.0.3	DNS	281	Standard query response
172.17.0.2	172.17.0.3	DNS	281	Standard query response

Figura 22: Tráfico víctima primera prueba

Efectivamente y como era de esperar, este es igual que el observado en el servidor DNS, salvo que, en este caso, las IP se encuentran intercambiadas.

Una vez analizado el tráfico por interfaces y haber observado el correcto funcionamiento en estas, podemos dar por concluida la prueba.

6.2 2 Atacantes / 1 Servidor DNSMASQ

Ya que la anterior prueba no ha conseguido satisfacer nuestros objetivos de llevar a cabo un ataque de denegación de servicio como consecuencia de no alcanzar un volumen de tráfico tal que fuera suficiente, se ha optado por aumentar el número de máquinas atacantes, de esta manera se pasa de realizar un ataque DoS a un ataque DDoS, ya que ahora este sí se encuentra distribuido entre varias máquinas.

En primer lugar, se va a mostrar un esquemático de lo que será la topología de nuestro nuevo entorno. Con ello se pretende mostrar de una forma más sencilla y visual los elementos que la componen.

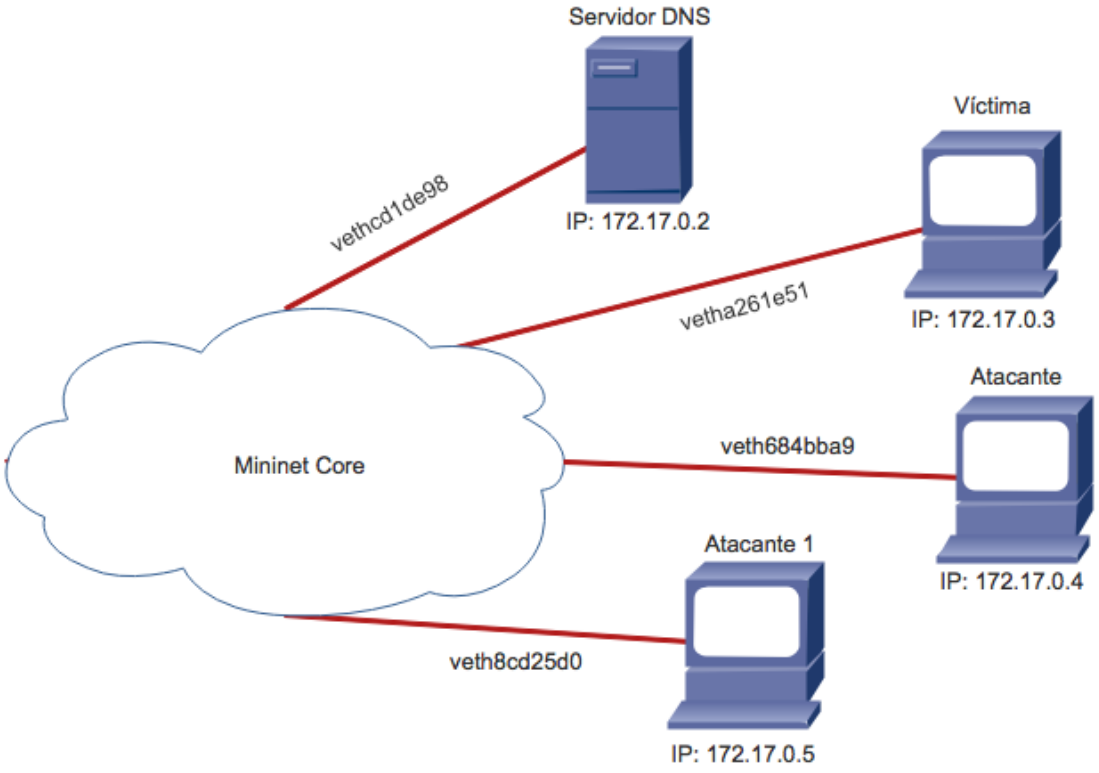


Figura 23: Entorno realización segunda prueba

El procedimiento a seguir en esta nueva prueba será muy similar al anteriormente llevado a cabo, por tanto, se pueden reutilizar elementos ya empleados como puede ser el fichero que contiene exportados los bytes correspondientes a la petición DNS a nuestro servidor. De nuevo, haremos uso de hping3 igualmente a como se procedió previamente, es decir, los parámetros que componen el comando serán los mismos y se aplicarán de la misma manera.

Si bien, en este caso todo aquello que correspondía al atacante y lo cual se llevaba a cabo en su equipo, ahora hemos de duplicarlo, es decir, se ha de llevar a cabo tanto en el atacante como el atacante1.

Primeramente, comprobamos el estado inicial de la maquina víctima del ataque, esta ha de encontrarse en una situación de reposo, en la cual la memoria RAM y la capacidad de cómputo de la CPU sean bajas.

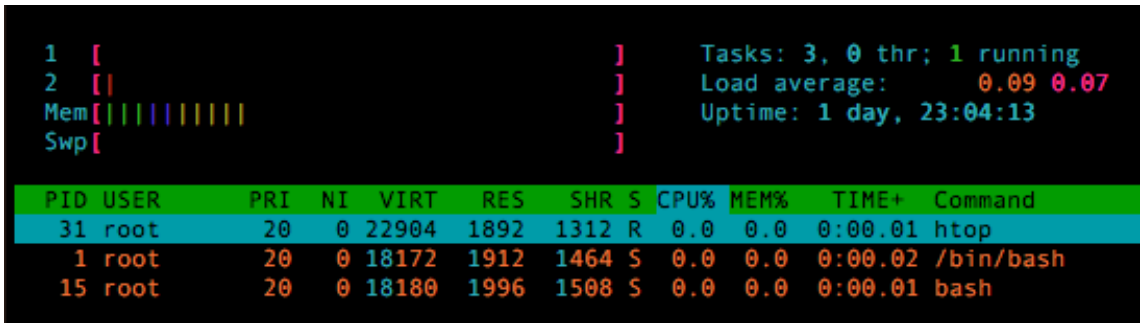


Figura 24:Rendimiento previo víctima segunda prueba

Los valores observados son correctos, y por tanto podemos proceder al siguiente paso, el cual consiste en llevar a cabo el ataque.

6.2.1 Desarrollo del ataque

Llegados a este punto, es el momento de hacer uso de hping3 para llevar a cabo el lanzamiento simultaneo del ataque desde ambos atacantes. Como ya se explicó en el caso anterior, hping3 consta de una serie de parámetros que podemos modificar a nuestras necesidades, ya sean frecuencia de envío de paquetes, tipo de protocolo que seguirán, el puerto de destino, etc… Como en este caso no se requiere ninguna modificación respecto al comando usado en la prueba anterior, usaremos para ambos atacantes el mismo comando, el cual se muestra a continuación:

hping3 --faster --udp -p 53 --spoof 172.17.0.3 --file query -d 28 172.17.0.2

Hacer hincapié de nuevo en el parámetro *–spoof*, el cual nos permite llevar a cabo la suplantación en la IP de origen y realizar de esta manera el intento de ataque de denegación de servicio.

Una vez se haga el lanzamiento conjunto del comando en ambos equipos que actuarán de atacantes, se ha de comprobar de nuevo el estado de la víctima para saber si esta se encuentra recibiendo el ataque y procesando los paquetes. Además, hemos de tener en cuenta de nuevo

el valor *Uptime*, para controlar así el tiempo durante el cual hemos estado llevando a cabo el ataque.

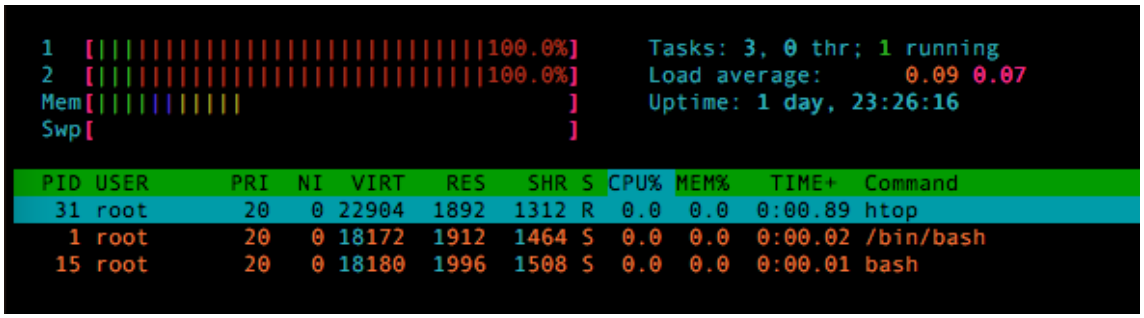


Figura 25: Estado inicial víctima segunda prueba

Como se puede apreciar en la imagen, el ataque se encuentra llevándose a cabo, ya que los niveles de cómputo de CPU en ambos núcleos han ascendido hasta su máximo valor. Esto es buen indicativo para poder llevar a cabo el ataque, ya que, sin niveles máximos de cómputo, no se puede alcanzar nuestro objetivo. Así como, el valor *Uptime* se sitúa en 1 día y 23:26:16 (horas/minutos/segundos).

Tras siete minutos observando el comportamiento de dichos valores, se decide detener el ataque al no estar consiguiendo los objetivos planteados. Como se puede observar en la imagen a continuación, el valor de la memoria RAM a lo largo de este lapso de tiempo no ha sufrido ninguna variación, siendo esto indicativo de que el ataque no está siendo efectivo.

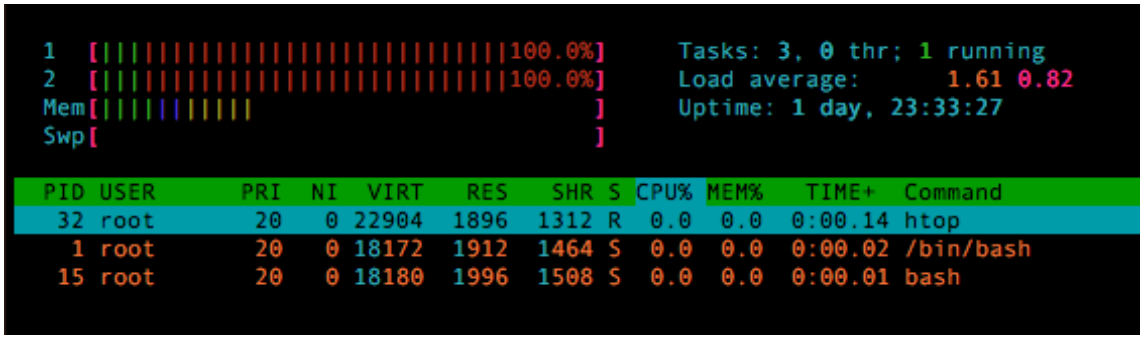


Figura 26: Estado final víctima segunda prueba

Como previamente se ha expuesto, debido a que nos encontramos trabajando sobre una virtualización, los recursos de cómputo en CPU y memoria RAM son compartidos, por ello hemos de ser consecuentes en cuanto si lo que se encuentra saturándose es realmente el equipo de la víctima o nuestro servidor DNS.

Por ello, pese a no haber conseguido una denegación de servicio a lo largo de esta prueba, podemos sacar una serie de conclusiones valiosas que nos serán de utilidad en futuras fases del

proyecto y las cuales nos permitirán obtener resultados veraces y no erróneos. Dicha conclusión principal, es la capacidad del servidor DNS de lidiar con el tráfico resultante de recibir dos ataques simultáneos, esto nos será de utilidad en caso de que necesitemos aumentar el número de servidores DNS, así como, de atacantes.

6.2.2 Análisis del tráfico

A continuación, se muestran diferentes capturas de Wireshark, en las cuales se muestra el tráfico de paquetes en las diferentes interfaces.

6.2.2.1 Tráfico interfaces atacantes

Ya que en este caso se tienen dos interfaces atacantes, las capturas de pantalla mostrarán también la interfaz desde la cual se está obteniendo dicha información. Las interfaces que se muestran se corresponden con las mostradas en la imagen de la topología del entorno.

A continuación, se mostrarán ambas capturas, y posteriormente se comentarán los resultados y el tráfico obtenidos en ellas. En primer lugar, se muestra la captura de la interfaz del atacante.

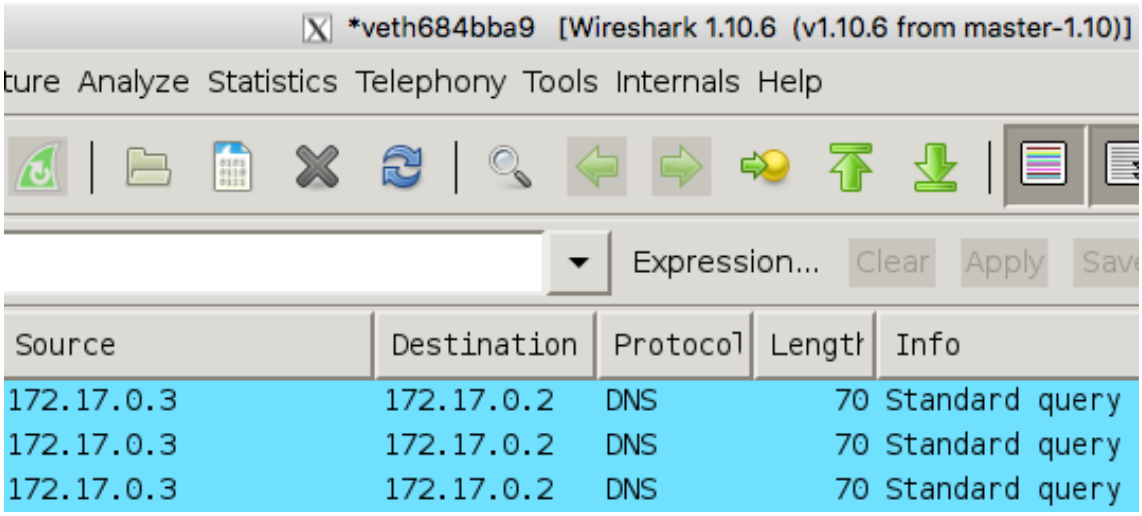


Figura 27: Tráfico atacante segunda prueba

En ella podemos apreciar claramente en la parte central superior de la imagen, en que interfaz nos encontramos capturando el tráfico. Tras haber mostrado ya la interfaz del atacante, es turno ahora de la interfaz correspondiente al atacante1.

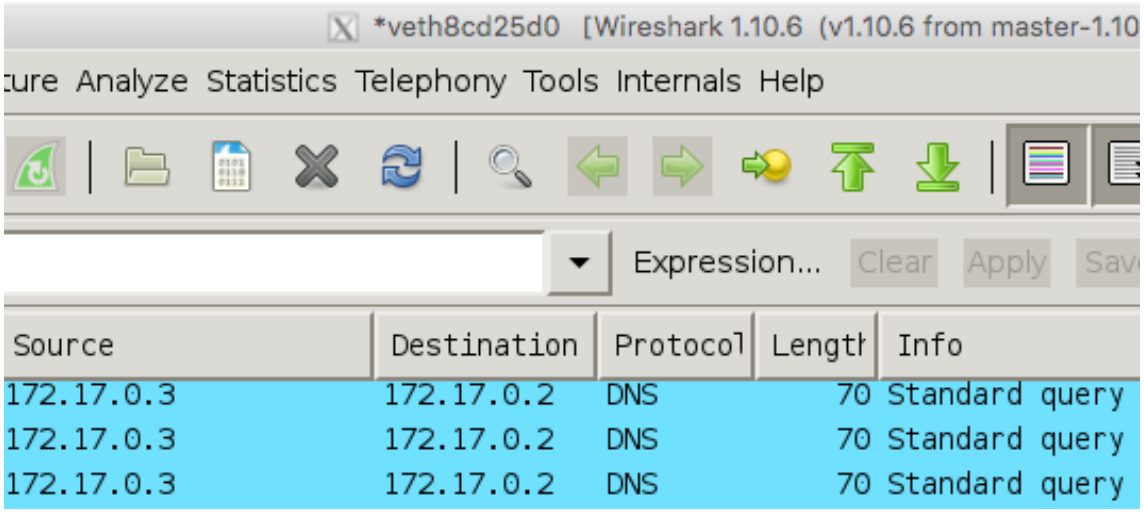


Figura 28: Tráfico atacante1 segunda prueba

En ambas imágenes podemos observar como el tráfico saliente posee la misma dirección IP de origen, esto es debido a la suplantación que se lleva a cabo gracias a hping3, siendo esta dirección la correspondiente de la víctima. Por ello, era necesario hacer la puntualización de las interfaces en este proceso de captura, para que se viese con claridad como desde ambas se obtiene el mismo tráfico saliente. Además, podemos observar como dichos paquetes salientes pertenecen al protocolo DNS, con carácter de petición, ya que la longitud de paquete que se observa es de 70 bytes.

En lo referente a estas capturas, el ataque discurre de manera correcta y por tanto podemos continuar.

6.2.2.2 Tráfico interfaz servidor DNS

En cuanto al tráfico a observar en esta interfaz, este ha de ser idéntico al del caso anteriormente analizado, ya que únicamente se tiene un servidor DNS. Pese a ello, se procede a observarlo, con el fin de saber si el intento de ataque estaba discurriendo de la manera adecuada.

Primeramente, se analizará el tráfico entrante al servidor con IP de origen la de la víctima e IP de destino la del servidor DNS. Para ello se empleará la expresión de filtrado del tráfico *ip.src==172.17.0.3 and ip.dst==172.17.0.*2.

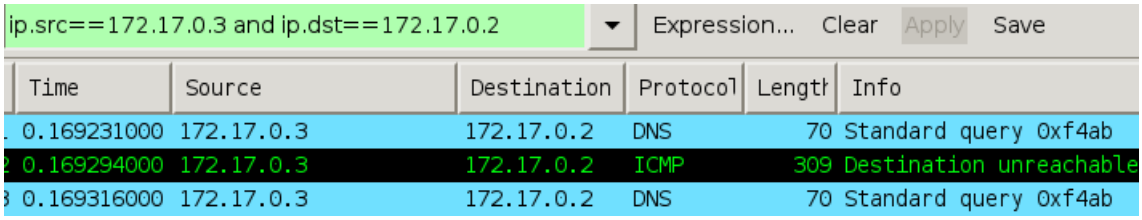


Figura 29: Tráfico DNS 1 segunda prueba

En ella se puede observar, como se obtiene el mismo resultado que en la prueba anterior con un único atacante. Esto como previamente se ha comentado, se debe al uso de un único servidor DNS, por lo que independientemente del número de atacantes que se tengan, el tráfico obtenido no en cuanto a volumen, si no a captura, será el mismo, ya que todos ellos llevarían a cabo la suplantación de la IP de origen y dirigen sus paquetes al mismo servidor DNS.

En lo relativo al tráfico saliente del servidor DNS con dirección IP de origen la del propio servidor y destino la de la víctima, comprobamos que este sea correcto y que todo discurre con normalidad.

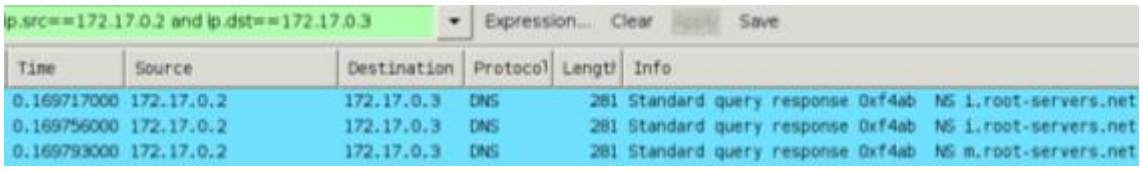


Figura 30: Tráfico DNS 2 segunda prueba

Como era de esperar, el tráfico saliente es correcto y las peticiones DNS se están procesando adecuadamente ya que se las respuestas a ellas se envían sin problema.

En cuanto al tráfico entre nuestro servidor DNS y los servidores de nombre que se ejecutan en segundo plano, en este caso no tenemos comunicación alguna entre ellos. Esto es debido a como anteriormente se ha explicado, a que el servidor DNS almacena en su memoria caché dicha información y por tanto ya no requiere de comunicarse con ellos para obtener dicha información.

6.2.2.3 Tráfico interfaz víctima

De nuevo, nos encontramos en la situación de que el tráfico que hemos de observar es idéntico respecto al de la prueba anteriormente realizada con un único atacante, así como, este ha de ser el mismo que el capturado anteriormente en el servidor DNS, únicamente con las direcciones IP intercambiadas entre sí.

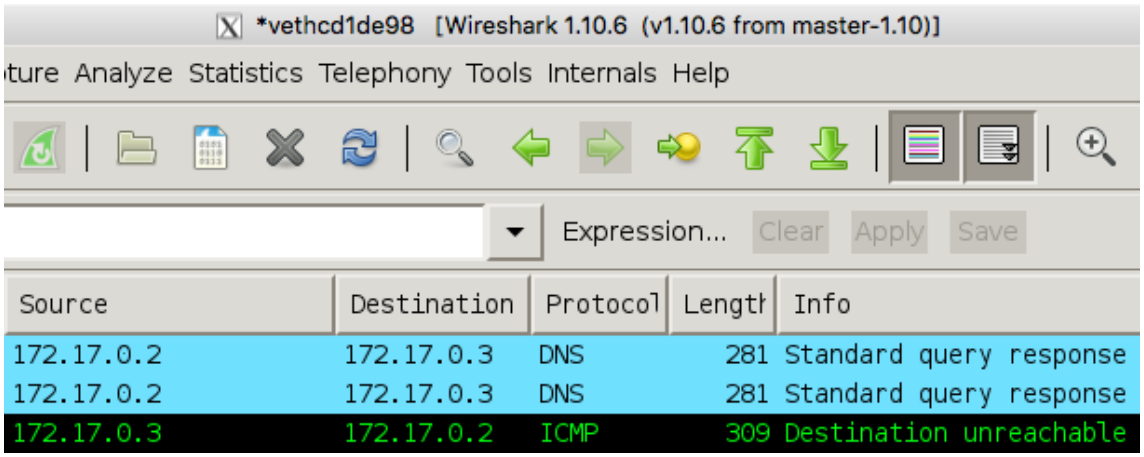


Figura 31: Tráfico víctima segunda prueba

Como se aprecia, solo se tiene tráfico entrante en forma de respuesta DNS proveniente del servidor, salvo aquellos paquetes de carácter ICMP, salientes del equipo de la víctima con destino el servidor DNS.

Por tanto, tras analizar el tráfico en las diferentes interfaces que componen nuestro entorno, podemos decir que dicho tráfico es correcto en todas ellas, si bien el ataque no ha sido efectivo, y no se ha logrado conseguir una denegación de servicio en la víctima.

6.3 3 Atacantes / 1 Servidor DNSMASQ

Nuevamente en la fase anterior del estudio no se logró alcanzar la denegación de servicio en la víctima, cierto es que los niveles de CPU aumentaron significativamente, si bien en ningún momento se llegó a conseguir que el equipo de la víctima se viese obligado a aumentar su uso de la memoria RAM. Por ello para para continuar el estudio, en esta siguiente fase se añadirá un nuevo atacante, de esta manera nuestro entorno contará con tres atacantes que realizarán peticiones al mismo servidor DNS.

Para que posteriormente en el análisis del tráfico de paquetes sea más sencillo, a continuación, se adjunta un esquemático de la red sobre la cual vamos a trabajar, especificando las diferentes interfaces que en esta nos encontramos.

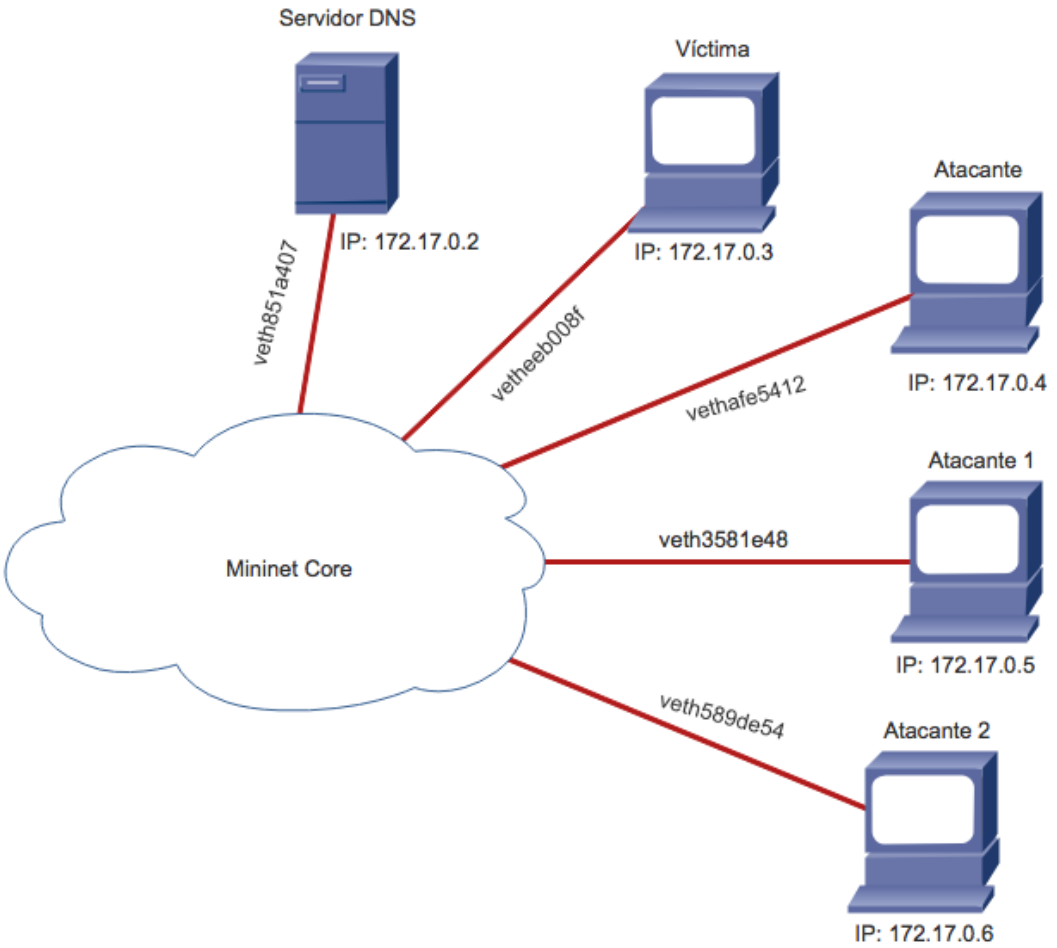


Figura 32: Entorno realización tercera prueba

Como en los casos anteriores, lo primero que comprobamos son los niveles de cómputo de CPU y de memoria RAM, para asegurarnos que ningún proceso se encuentra ejecutándose y que todo es correcto antes de iniciar el ataque.

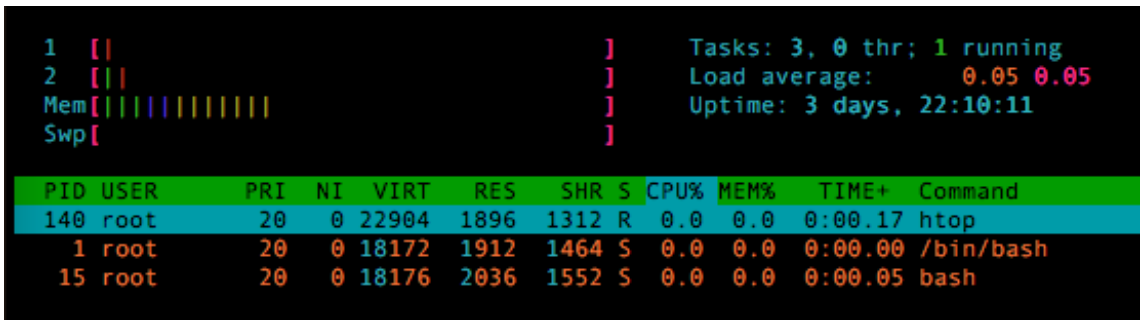


Figura 33: Rendimiento previo víctima tercera prueba

En la imagen se aprecia como los valores de cómputo de la CPU son bajos, ya que ningún proceso se encuentra ejecutándose, si bien en el caso de la memoria RAM el valor es un poco mayor dado que lo que htop nos muestra es el conjunto de la memoria de la virtualización, y ese pequeño aumento se debe a la presencia del nuevo atacante. Por ello, y para complementar

dicha imagen, haremos uso del comando Docker Stats, para así observar el porcentaje de recursos que se encuentra consumiendo cada contenedor. A continuación, se muestran tanto el comando empleado para ello, como la imagen con los resultados obtenidos.

docker stats victima atacante atacante1 atacante2 dns

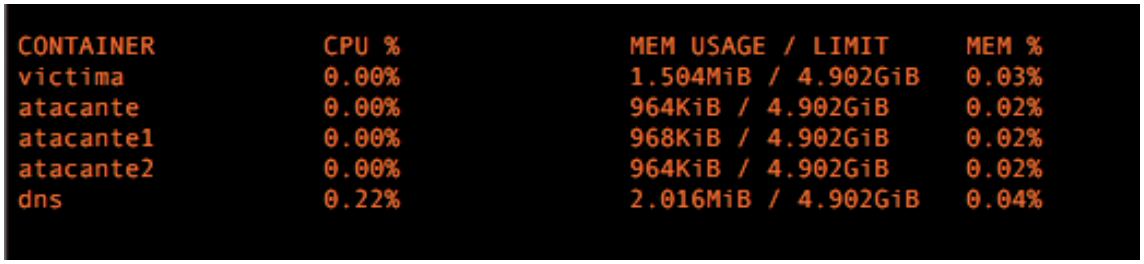


Figura 34: Valores iniciales CPU-RAM tercera prueba

Como se puede apreciar, el consumo de recursos es bastante inferior a lo que el comando htop podría darnos a entender, ya que el total de memoria RAM empleado entre los cinco contenedores no llega al 1% del total. Una vez hemos comprobado estos valores, podemos proceder a la ejecución del ataque.

6.3.1 Desarrollo del ataque

De nuevo, el procedimiento será idéntico al ejecutado en las dos fases anteriores, si bien el único cambio que vamos a encontrar llegados a este punto, es la ejecución del ataque simultáneamente desde los tres atacantes que se tienen. Para ello de nuevo haremos uso del comando hping3 e inmediatamente después comprobaremos los valores *Uptime* y consumos de CPU y memoria RAM para asegurarnos de que el ataque ha comenzado a ejecutarse.

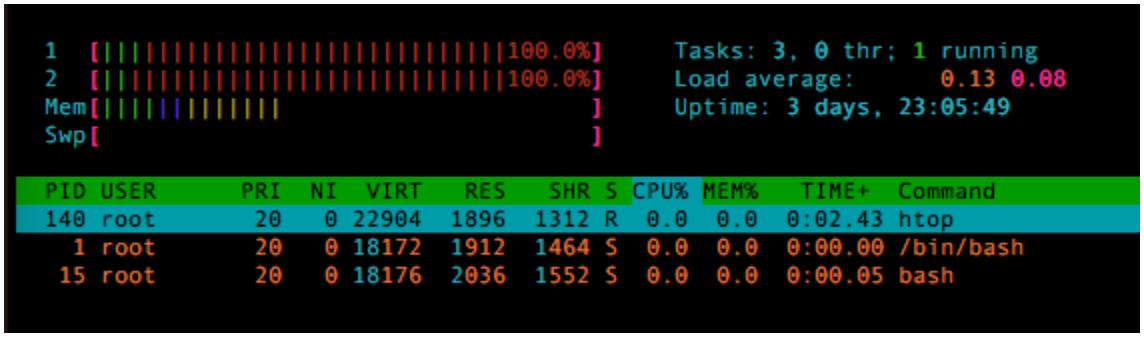


Figura 35: Estado inicial víctima tercera prueba

Como podemos apreciar, el ataque ha sido lanzado ya que los indicadores de cómputo de CPU se encuentran registrando sus máximos valores. El valor de *Uptime* registrado es de 3 días y 23:05:49 (horas/minutos/segundos).

Adicionalmente, vamos a observar los valores que nos devuelve el comando docker stats, de esta manera queremos observar el nivel de computo que poseen el servidor DNS y el equipo de la víctima, ya que se sospecha que pueda existir una sobrecarga en el servidor DNS por el volumen de paquetes entrantes, y que por tanto este no sea capaz de procesar todos ellos, haciendo por tanto nulo el ataque.

Figura 36: Valores CPU, RAM ataque tercera prueba

Como se puede apreciar, en los instantes iniciales las máquinas atacantes se encuentran empleando grandes cantidades de sus recursos disponibles de CPU, mientras que el servidor DNS y la víctima rondan valores muy bajos, casi ínfimos. Es momento ahora de analizar cómo evolucionan estos parámetros, y si la máquina víctima del ataque comienza a emplear grandes cantidades de memoria RAM.

Transcurridos varios minutos, observamos como el porcentaje de CPU empleado tanto por la víctima como por el servidor DNS son muy bajos en comparación con los atacantes, así como, los niveles de uso de memoria RAM se mantienen estables en sus valores iniciales sin observarse grandes modificaciones. Esto indica que el ataque está siendo fallido y que no se está consiguiendo lograr la denegación de servicio esperada en la víctima.

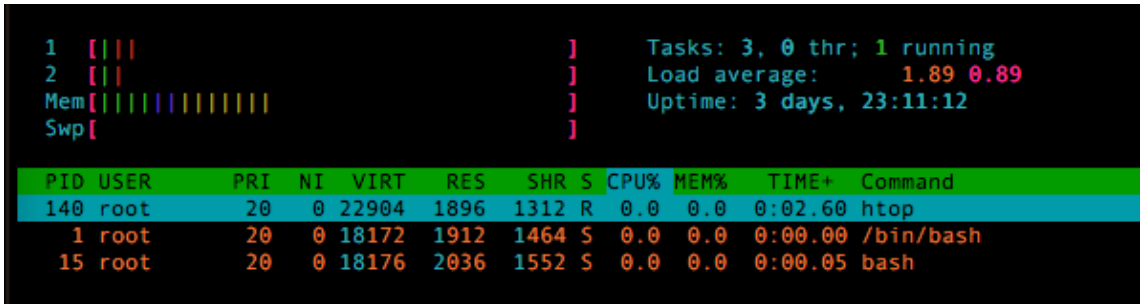


Figura 37: Estado final víctima tercera prueba

Como podemos apreciar en la imagen, se observa como el conjunto de la memoria no ha aumentado su valor tras seis minutos de desarrollo del ataque (valor *Uptime* de 3 días 23:11:12 (horas/minutos/segundos)).

Un parámetro de interés para intentar comprender el porqué de esta situación es el campo NET I/O del comando *docker stats*, y cual procedemos a analizar a continuación:

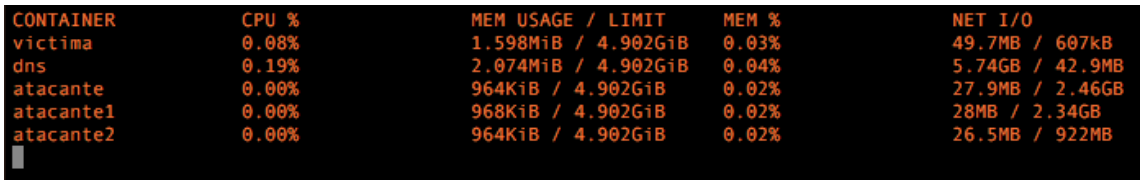


Figura 38: Valores NET I/O final tercera prueba

Si observamos detenidamente dicho parámetro, podremos analizar la relación tráfico entrante/saliente que ha registrado el contenedor, lo cual nos permitirá saber el punto en el cual el ataque no se desarrolla de manera adecuada.

- En cuanto a los atacantes, observamos como la diferencia entre los valores de entrada y salida son realmente diferenciados, valores de megabytes en lo que al tráfico entrante se refiere, mientras que el saliente es del orden de gigabytes. Esto es algo de esperar ya que el tráfico registrado en sus interfaces es principalmente el destinado a la realización del ataque, con la IP de origen modificada por la de la víctima, por lo que estos no reciben respuesta alguna a dichos paquetes.
- Observando los valores correspondientes al servidor DNS, apreciamos las primeras incongruencias que derivan en el mal desarrollo del ataque y por tanto que no consigamos su realización. Como se puede apreciar, el valor de tráfico entrante es de 5.74 GB mientras que el saliente posee un valor de 42.9 MB. Esto refleja como el servidor DNS no es capaz de lidiar con todo el tráfico entrante y transformarlo en saliente, siendo imposible dados estos valores, la posibilidad de llevar a cabo la denegación de servicio en la víctima.
- En lo que se refiere a la víctima, el tráfico entrante ha de tener un valor similar y cercano al saliente del servidor DNS, hecho que se verifica, ya que el valor registrado es de 49.7 MB. Por tanto, es lógico que no se logre la denegación de servicio, ya que con esos valores de tráfico entrante es imposible colapsar el equipo.

Llegados a este punto, podemos afirmar que el problema se está dando en el servidor DNS, ya que, como previamente se ha comentado, este no es capaz de convertir todas las peticiones entrantes desde los atacantes, en respuestas salientes para la víctima.

6.3.2 Análisis del tráfico

A diferencia de lo incluido en los casos anteriores en este apartado, en esta ocasión no se mostrará ninguna captura del tráfico recogido en las diferentes interfaces, si bien, se llevará a cabo una breve explicación del proceso llevado a cabo para la verificación de que realmente existe comunicación entre los atacantes y la víctima, y que dicha comunicación se llevaba a cabo de manera correcta.

Dado que el problema que se ha encontrado reside en que el servidor DNS no cursa todo el tráfico entrante, se decide llevar a cabo pruebas individuales con cada uno de los atacantes con el fin de verificar la comunicación entre estos y la víctima. Para ello haremos uso de nuevo de la herramienta Wireshark.

Tras llevar a cabo las pruebas con los tres atacantes, podemos obtener varias conclusiones que derivan en la forma de proceder en la continuación del estudio. Dichas pruebas verificaron que la comunicación en correcta, si bien como se preveía, el número de paquetes cursados por el servidor DNS hacia la víctima es muy inferior al número de paquetes entrantes, por ejemplo, en la prueba realizada con el primer atacante, de 18.168 paquetes enviados por el atacante, únicamente 689 fueron cursados a la víctima. Por tanto, para continuar con el estudio, se decide incluir un nuevo servidor DNS y reducir un atacante, de tal forma que se tengan dos atacantes y dos servidores DNS, de esta manera tratamos de dividir el tráfico entre estos dos servidores y observaremos si de esta manera se consigue un mayor tráfico entrante en la víctima.

6.4 2 Atacantes / 2 Servidores DNSMASQ

En esta nueva fase del estudio se incluirá un nuevo servidor DNS y se eliminará un atacante, lo cual implica llevar a cabo una serie de cambios en la arquitectura de la red, ya que recordemos, al inicio del estudio se asignaron una serie de valores que se traducían en la capacidad máxima de cómputo de CPU disponible para cada contenedor. Por ello, hemos de realizar un reajuste en dichos valores previamente asignados, de tal forma que podamos incluir el nuevo servidor, con una capacidad de cómputo similar a su semejante.

La nueva asignación de valores será de un 25% para cada uno de los servidores DNS, un 25% para la víctima, y de un 6% para cada uno de los atacantes. Esto se traduce en una asignación de 87% de la capacidad de cómputo de la virtualización. Con la incorporación del nuevo servidor DNS y la asignación de estos nuevos valores se pretende conseguir la denegación de servicio en la víctima. La razón de disminuir los porcentajes de CPU asignados a cada contenedor se debe a que los servidores DNS al ser de tipo dnsmasq, han de realizar consultas a un servidor de nombre

externo, el cual se encuentra fuera de lo que es el propio contenedor, por tanto, hemos de dejar un margen de cómputo para que se procesen dichas peticiones, tratando de evitar así que el fallo pueda residir en ese punto del proceso.

Para la creación del segundo contenedor DNS, ya que el puerto 53 de la dirección 172.17.0.1 ya se encontraba asignado al primer servidor, se buscó un número de puerto el cual estuviese libre y que por tanto pudiera ser utilizado para cursar nuestro tráfico, dicho puerto elegido es el número 100. A continuación, se muestra la creación de ambos contenedores DNS:

docker run --detach --name dns -c 256 --publish 172.17.0.1:53:53/udp --volume /var/run/docker.sock:/var/run/docker.sock jderusse/dns-gen

docker run --detach --name dns1 -c 256 --publish 172.17.0.1:100:53/udp --volume /var/run/docker.sock:/var/run/docker.sock jderusse/dns-gen

De esta manera ya se tienen los dos servidores DNS lanzados y listos para ser usados. Después de estos, hemos de ejecutar los contenedores de la víctima y de los dos atacantes:

docker run -itd -c 256 --name victima sameersbn/ubuntu

docker run -itd -c 62 --name atacante sameersbn/ubuntu

Por tanto, ya tendríamos el nuevo entorno creado y listo para llevar a cabo la prueba del ataque. A continuación, se muestra el esquemático del nuevo entorno a evaluar:

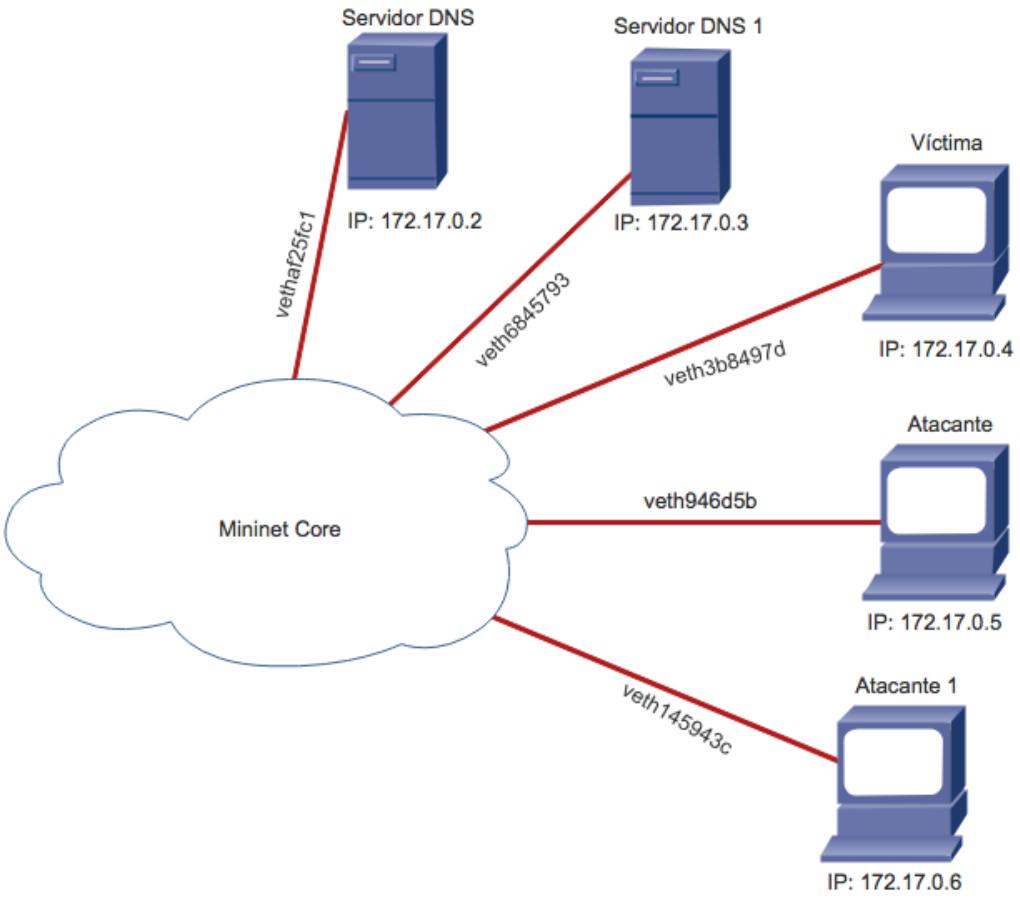


Figura 39: Entorno realización cuarta prueba

Una vez llegados a este punto, hemos de comprobar que todo funcione correctamente y con el fin de evitar hallar errores más adelante en el proceso, se deciden llevar a cabo diferentes pruebas para verificar el correcto funcionamiento. Primeramente, comprobaremos los valores NET I/O del comando *docker stats*, para comprobar los valores iniciales y después poder compararlos una vez hayamos realizado la prueba.

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %               NET I/O
victima             0.00%               956KiB / 4.902GiB     0.02%               23.9MB / 481kB
dns                 0.17%               2.043MiB / 4.902GiB   0.04%               757kB / 747kB
dns1                0.24%               2.023MiB / 4.902GiB   0.04%               204B / 0B
atacante            0.00%               960KiB / 4.902GiB     0.02%               26.1MB / 391kB
atacante1           0.00%               968KiB / 4.902GiB     0.02%               26.3MB / 518kB
```

Figura 40: Valores CPU, RAM, NET I/O previos cuarta prueba

Como se puede apreciar, los valores son ínfimos respecto a los recogidos al finalizar la prueba anterior, esto es debido a la nueva creación de estos contenedores. El poco tráfico que se ha recogido se debe principalmente a la instalación de las diferentes herramientas necesarias tanto en los equipos de los atacantes como de la víctima.

En segundo lugar, dado que el segundo servidor DNS se encuentra empleando un puerto no habitual, analizaremos si los servidores DNS son capaces de responder a una simple petición y devolvernos los servidores de nombre asociados. Únicamente se incluye imagen de la petición al segundo servidor (dns1), aquel que emplea el puerto 100, ya que es el que pudiera dar alguna clase de problema.

```
root@4e59e6e1ae74:/# dig @172.17.0.3

; <<>> DiG 9.9.5-3ubuntu0.15-Ubuntu <<>> @172.17.0.3
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57586
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 1280
;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       5       IN      NS      i.root-servers.net.
.                       5       IN      NS      b.root-servers.net.
.                       5       IN      NS      g.root-servers.net.
.                       5       IN      NS      h.root-servers.net.
.                       5       IN      NS      k.root-servers.net.
.                       5       IN      NS      f.root-servers.net.
.                       5       IN      NS      l.root-servers.net.
.                       5       IN      NS      j.root-servers.net.
.                       5       IN      NS      d.root-servers.net.
.                       5       IN      NS      c.root-servers.net.
.                       5       IN      NS      e.root-servers.net.
.                       5       IN      NS      a.root-servers.net.
.                       5       IN      NS      m.root-servers.net.

;; Query time: 54 msec
;; SERVER: 172.17.0.3#53(172.17.0.3)
;; WHEN: Wed Sep 13 14:52:18 UTC 2017
;; MSG SIZE  rcvd: 239
```

Figura 41: Consulta DNS nuevo servidor cuarta prueba

Como podemos observar, el servidor responde correctamente a la petición, devolviendo los servidores de nombre asociados, además en dicha respuesta se especifica que no ha ocurrido ningún error, y por tanto podemos proseguir con el estudio.

Momento ahora de obtener la petición DNS correspondiente al nuevo servidor, la cual posteriormente será empleada para el desarrollo del ataque. Como se hizo en el primer caso de estudio, emplearemos Wireshark para capturar dicho paquete y posteriormente exportarlo para su posterior uso. Únicamente es necesario realizarlo con este servidor (dns1), ya que con el

anterior (dns) al mantener la misma dirección IP, podemos reutilizar la consulta exportada al inicio del estudio. [48]

Source	Destination	Protocol	Length	Info
172.17.0.6	172.17.0.3	DNS	70	Standard query 0x3cbe
172.17.0.3	172.17.0.6	DNS	281	Standard query response

Figura 42: Consulta DNS nuevo servidor Wireshark cuarta prueba

Como se puede comprobar, el servidor ha respondido correctamente a la petición realizada, y por tanto podemos decir que funciona correctamente pese a estar alojado en un puerto diferente al 53, el cuál es el puerto por defecto.

Por tanto, todo es correcto y llegados a este punto podemos proceder a la realización del ataque y observar si este se ejecuta correctamente o no. Si bien antes de llevarlo a cabo, analizamos los niveles de cómputo y de memoria de los diferentes contenedores de tal forma que comprobemos que todo se encuentra en valores correctos.

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %
victima             0.00%               956KiB / 4.902GiB     0.02%
dns                 0.14%               2.953MiB / 4.902GiB   0.06%
dns1                0.10%               2.082MiB / 4.902GiB   0.04%
atacante            0.00%               964KiB / 4.902GiB     0.02%
atacante1           0.00%               968KiB / 4.902GiB     0.02%
```

Figura 43: Valores CPU, RAM previos cuarta prueba

Los valores son correctos y por tanto procedemos a la ejecución del ataque.

6.4.1 Desarrollo del ataque

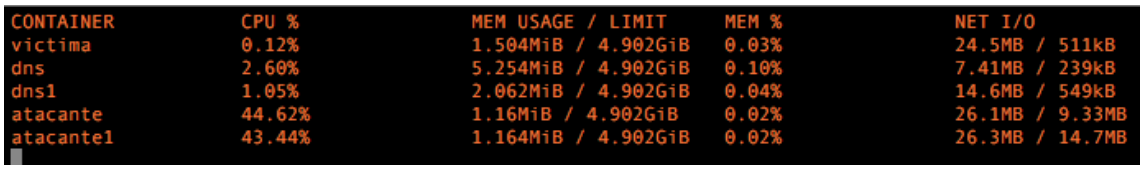
El desarrollo del ataque seguirá el mismo orden que en las pruebas anteriormente realizadas: en primer lugar, lanzaremos el comando *hping3* simultáneamente desde ambos atacantes, cada uno de ellos dirigido a un servidor DNS; inmediatamente después analizaremos si el ataque ha comenzado a ejecutarse; y finalmente transcurrido un tiempo observaremos si este se ha desarrollado correctamente consiguiendo nuestro objetivo, o si de nuevo se fracasa en el intento.

Ya que en esta ocasión se han modificado los servidores DNS y atacantes, a continuación, se incluyen los comandos *hping3* empleados, de tal forma que se aprecie como ambos se ejecutan para realizar peticiones a dos servidores DNS diferentes:

hping3 --faster --udp -p 53 --spoof 172.17.0.4 --file query -d 28 172.17.0.2

hping3 --faster --udp -p 53 --spoof 172.17.0.4 --file query1 -d 28 172.17.0.3

Como podemos observar, ambos poseen la misma dirección a falsificar, la de la víctima, si bien en la dirección del servidor DNS a realizar la petición esta es diferente, el primer comando se ejecuta dirigiéndose a dns, mientras que el segundo se dirige a dns1. Además, se aprecia como el fichero empleado para llevar a cabo la consulta también difiere de un comando al otro. En cuanto al puerto de destino, en ambos casos es el 53, esto no se trata de un error, ya que internamente ambos servidores emplean el puerto 53 como el designado para el procesado de peticiones DNS. Si bien, en el caso del servidor dns1, el puerto público al cual se mapea es el número 100, de ahí que anteriormente se le asignara dicho valor.

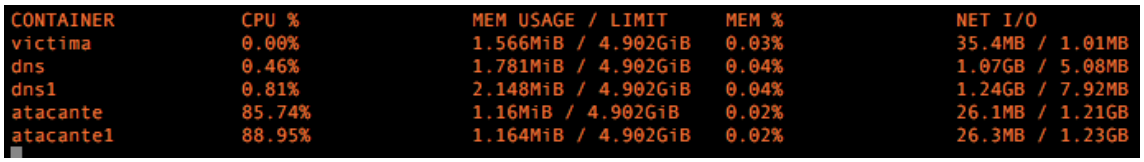


CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O
victima	0.12%	1.504MiB / 4.902GiB	0.03%	24.5MB / 511kB
dns	2.60%	5.254MiB / 4.902GiB	0.10%	7.41MB / 239kB
dns1	1.05%	2.062MiB / 4.902GiB	0.04%	14.6MB / 549kB
atacante	44.62%	1.16MiB / 4.902GiB	0.02%	26.1MB / 9.33MB
atacante1	43.44%	1.164MiB / 4.902GiB	0.02%	26.3MB / 14.7MB

Figura 44: Valores CPU, RAM, NET I/O iniciales cuarta prueba

En la imagen se aprecia como el ataque ha comenzado a ejecutarse, los valores de cómputo en los atacantes rondan el 45%, lo que significa que han empezado a enviar peticiones a los servidores DNS, así como, se aprecia como el tráfico saliente desde estos ha aumentado rápidamente sus valores.

Transcurridos unos minutos observamos que el ataque no está siendo efectivo, de nuevo la víctima se encuentra en valores ínfimos tanto en cómputo de CPU como en uso de memoria, al igual que los servidores DNS, como se puede apreciar en la imagen que se muestra a continuación.



CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O
victima	0.00%	1.566MiB / 4.902GiB	0.03%	35.4MB / 1.01MB
dns	0.46%	1.781MiB / 4.902GiB	0.04%	1.07GB / 5.08MB
dns1	0.81%	2.148MiB / 4.902GiB	0.04%	1.24GB / 7.92MB
atacante	85.74%	1.16MiB / 4.902GiB	0.02%	26.1MB / 1.21GB
atacante1	88.95%	1.164MiB / 4.902GiB	0.02%	26.3MB / 1.23GB

Figura 45: Valores CPU, RAM, NET I/O finales cuarta prueba

Adicionalmente, en la imagen se aprecia como el tráfico saliente de los atacantes no se encuentra siendo cursado por los servidores DNS, de nuevo nos encontramos ante el mismo problema que el caso anterior. Los atacantes rondan valores de 1.2GB de tráfico saliente, valores similares al entrante en los servidores, si bien si observamos el tráfico saliente de estos es del

orden de 7MB, valores ínfimos comparados con el entrante. Como consecuencia, el tráfico entrante en la víctima es mínimo, de tal forma que lograr la denegación de servicio es imposible.

6.4.2 Análisis del tráfico

Al igual que sucedió en el caso anterior, encontrándonos ante esta situación, pese a haber verificado la comunicación previamente antes de realizar el ataque, se decide volver a comprobar el tráfico de paquetes entre atacante, servidor DNS y víctima, para los dos pares atacante-servidor que se tienen. Como cabía de esperar, en ambos casos la comunicación es correcta:

- El tráfico saliente del atacante posee la IP de origen falsificada por la de la víctima y con IP de destino la de su correspondiente servidor DNS.
- En el servidor, se aprecian los paquetes entrantes y como el servidor recurre a los servidores de nombre que se ejecutan en segundo plano, propio de un servidor dnsmasq, y finalmente como envía estos paquetes a la víctima.
- En cuanto a lo que la víctima se refiere, se observan los paquetes entrantes con IP de origen la de nuestro servidor DNS, y con IP de destino la propia de la víctima.

Si bien es cierto que el número de paquetes entrantes en la víctima es menor que el número saliente en el atacante, la diferencia es bastante inferior a la que se observa transcurridos varios minutos de comunicación. Por tanto, cuanto mayor es el lapso de tiempo que el ataque se encuentra activo, mayor es la diferencia entre el número de paquetes salientes del atacante y entrantes en la víctima.

Llegados a este punto, se barajaron diferentes conclusiones de porqué el ataque no se realizaba correctamente:

- La primera de ellas fue deducir que, al tratarse de una virtualización, el propio sistema de docker al detectar un elevado volumen de tráfico con carácter reflexivo en un servidor DNS, no permitiese llevar a cabo el ataque, desechando por tanto un gran número de paquetes en el servidor.
- La segunda de ellas fue reflexionar acerca del tipo de servidor DNS empleado, como previamente se ha expuesto en el punto 5.2.2.1, el servidor idóneo para la realización del experimento era un servidor de tipo BIND, si bien por diferentes razones ya comentadas, se optó por el de tipo dnsmasq. Por ello se ha llegado a pensar que puede ser debido al tipo de servidor, y que por tanto usando el originalmente establecido (BIND) se lograra llevar a cabo con éxito el ataque.

Por ello se sopesaron ambas opciones y se decidió volver al servidor BIND, tratar de configurarlo correctamente y de esta manera llevar a cabo una nueva fase en el estudio.

6.5 3 Atacantes / 1 Servidor DNS BIND

La última fase del estudio consistirá en llevar a cabo el ataque desde tres máquinas simultáneamente, todas ellas empleando el mismo servidor DNS, en esta ocasión de tipo BIND, a diferencia de las anteriores pruebas en las cuales el servidor era de tipo dnsmasq. La motivación principal del desarrollo de esta última fase es clarificar si el hecho de no haber conseguido previamente resultados satisfactorios en cuanto al desarrollo del ataque se refiere, se debe a la propia virtualización, o al tipo de servidores DNS empleados.

Siguiendo la metodología empleada en el resto de fases, se muestra una imagen con el esquemático del entorno en cual desarrollaremos las pruebas:

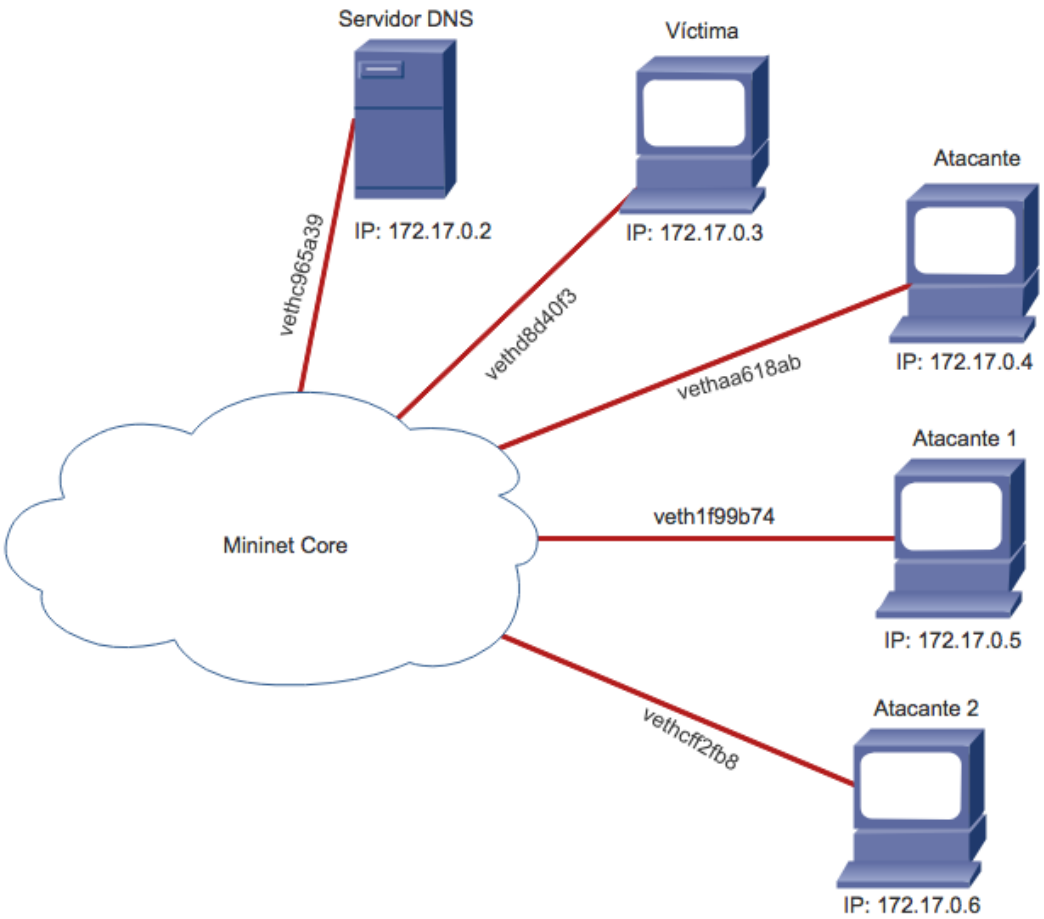


Figura 46: Entorno realización quinta prueba

Para el montaje del nuevo servidor DNS se empleó el comando que se muestra a continuación:

docker run --name bind -d --restart=always --publish 53:53/tcp --publish 172.17.0.1:53:53/udp --publish 10000:10000/tcp --volume /srv/docker/bind:/data sameersbn/bind:9.9.5-20170626

En este se observa como para el tráfico udp en el puerto 53 se mapea dicha dirección con la propia de docker.

Dado que se trata de un nuevo servidor, hemos de exportar nuevamente en un paquete la petición DNS que los atacantes emplearan para el desarrollo del ataque. Para ello haremos uso de Wireshark, y en esta ocasión la consulta a realizar pasa por pedir la dirección IP correspondiente al servidor de nombre asociado a nuestro servidor DNS. Tras mostrar la petición, se llevarán a cabo una serie de comentarios aclarando los parámetros del comando.

dig A ns.midominio.privado @172.17.0.2

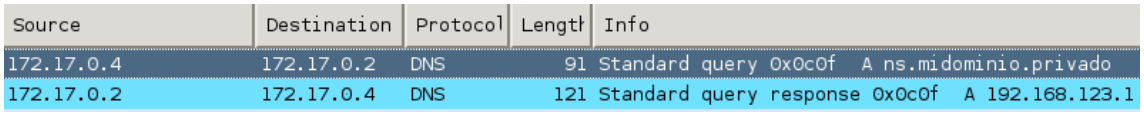
- *ns.midominio.privado* se corresponde con una zona creada internamente del servidor DNS, esta zona posee una dirección IP propia la cual queremos conocer, lo cual es posible gracias al uso del parámetro *A*.
- *@172.17.0.2* se corresponde con la dirección IP propia del servidor DNS al cual realizamos la consulta.

```
[root@4184a4470172:/# dig A ns.midominio.privado @172.17.0.2

; <<>> DiG 9.9.5-3ubuntu0.15-Ubuntu <<>> A ns.midominio.privado @172.17.0.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3087
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns.midominio.privado.          IN      A

;; ANSWER SECTION:
ns.midominio.privado.   86400   IN      A       192.168.123.1

;; AUTHORITY SECTION:
midominio.privado.      86400   IN      NS      ns.midominio.privado.

;; Query time: 0 msec
;; SERVER: 172.17.0.2#53(172.17.0.2)
;; WHEN: Thu Sep 14 15:48:45 UTC 2017
;; MSG SIZE  rcvd: 79
```

Figura 47: Consulta DNS quinta prueba

Como se puede apreciar en la imagen, el comando nos devuelve la dirección 192.168.123.1, que es la correspondiente al servidor de nombre asociado a la zona midominio.privado.

Momento ahora de mostrar la captura de dicha petición por medio de Wireshark, donde se observa la amplificación deseada para poder llevar a cabo el ataque. [48]



Source	Destination	Protocol	Length	Info
172.17.0.4	172.17.0.2	DNS	91	Standard query 0x0c0f A ns.midominio.privado
172.17.0.2	172.17.0.4	DNS	121	Standard query response 0x0c0f A 192.168.123.1

Figura 48: Consulta DNS Wireshark quinta prueba

Relativo de la imagen, comentar un par de cosas relevantes:

- La amplificación producida en este caso es menor dado que la consulta es más específica, y por tanto la respuesta no es tan grande como las anteriormente obtenidas. Si bien, aun así, existe amplificación y por tanto nos es válido para llevar a cabo la prueba. El hecho de que en esta ocasión la petición tenga que ser más concreta, es que el servidor carece de forwarders, y por tanto no responde a una petición genérica.
- Otro elemento a destacar es el campo *Info de ambos* paquetes, en el cual se aprecia cual es la consulta y la respuesta respectivamente.

Llegados a este punto es momento de comenzar el ataque y observar los resultados que se obtienen.

6.5.1 Desarrollo del ataque

Dado que la consulta es diferente, en primer lugar, hemos de verificar el tamaño del fichero exportado, ya que esto es necesario para el correcto lanzamiento del ataque, siendo este uno de los parámetros a incluir. Tras comprobarlo, se observa que este ha aumentado significativamente su tamaño, si bien esto no nos debe preocupar, salvo por lo previamente mencionado. Por tanto, el comando a ejecutar simultáneamente desde los tres atacantes será:

hping3 --faster --udp -p 53 --spoof 172.17.0.3 --file query -d 49 172.17.0.2

Antes del lanzamiento, comprobamos tanto los valores de cómputo de CPU como la memoria RAM que se encuentra siendo empleada, así como, los valores del tráfico entrante y saliente (NET I/O).

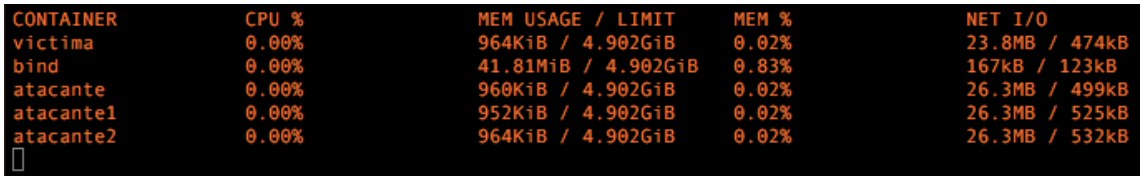


Figura 49: Valores CPU, RAM, NET I/O previos quinta prueba

Como se puede apreciar los valores son correctos y no se observa ninguna anomalía. Los valores referentes al tráfico se atribuyen principalmente a la instalación de las herramientas necesarias, además de a las consultas DNS realizadas para la obtención del fichero exportado.

Por tanto, podemos proceder ya al lanzamiento del ataque. Inmediatamente después de esto, se comprobará el valor *Uptime*, de tal forma que se tenga una referencia temporal de la ejecución del ataque a lo largo del tiempo.

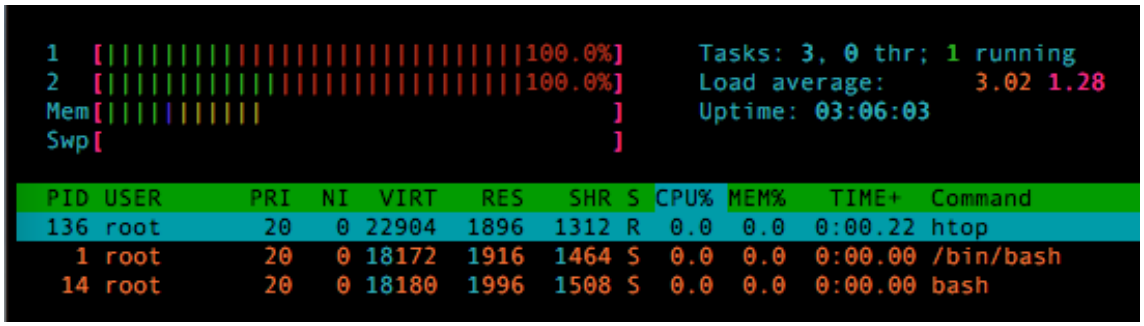


Figura 50: Valor Uptime inicial quinta prueba

Transcurridos varios minutos, observamos que el ataque no está siendo efectivo, si bien en esta ocasión los resultados son más esperanzadores que en las anteriores pruebas, ya que, en este caso, el servidor DNS si se encuentra cursando todo el tráfico entrante, como se puede observar en la imagen a continuación, obtenida durante el desarrollo del ataque.

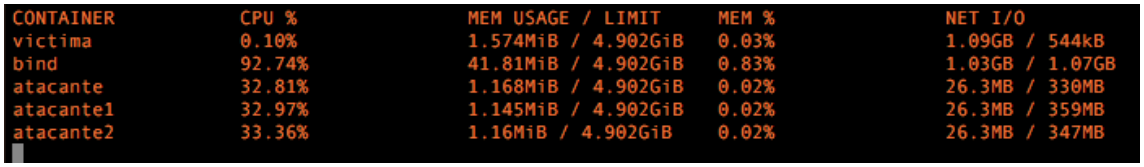


Figura 51: Valores CPU, RAM, NET I/O finales quinta prueba

Como se observa, la suma del tráfico saliente desde los tres atacantes es prácticamente el tráfico entrante en el servidor DNS, y a su vez, el tráfico saliente del servidor, se corresponde con el entrante en la víctima, siendo este de un valor superior debido a la amplificación que sufre. Esto denota que el ataque se encuentra realizándose correctamente, si bien, la amplificación de los paquetes no es suficiente para llevar a cabo la denegación de servicio con únicamente tres

atacantes. Otro enfoque sería que se requieren de más máquinas atacantes para poder desarrollar definitivamente la denegación de servicio en la víctima.

Por ello se decide probar una nueva consulta DNS en la cual la relación tamaño petición/respuesta sea bastante mayor, dicha petición consiste en vez de preguntar por la IP correspondiente al servidor de nombre, se pregunta por la IP del servidor de mail, la cual será la misma, pero se obtiene una relación mucho mayor, ya que la respuesta es mucho más completa, mientras que la petición se mantiene similar. A continuación, se muestra tanto la petición como la respuesta obtenida:

dig A mail.midominio.privado @172.17.0.2

```
[root@4184a4470172:/# dig A mail.midominio.privado @172.17.0.2

; <<>> DiG 9.9.5-3ubuntu0.15-Ubuntu <<>> A mail.midominio.privado @172.17.0.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30974
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;mail.midominio.privado.                IN      A

;; ANSWER SECTION:
mail.midominio.privado. 86400   IN      A       192.168.123.2

;; AUTHORITY SECTION:
midominio.privado.      86400   IN      NS      ns.midominio.privado.

;; ADDITIONAL SECTION:
ns.midominio.privado.   86400   IN      A       192.168.123.1

;; Query time: 0 msec
;; SERVER: 172.17.0.2#53(172.17.0.2)
;; WHEN: Thu Sep 14 17:05:33 UTC 2017
;; MSG SIZE  rcvd: 100
```

Figura 52: Consulta DNS 2 quinta prueba

Una vez hemos observado como dicha respuesta obtenida realmente es más completa, es momento de analizar la relación petición/respuesta existente, y observar la amplificación que se produce.

Source	Destination	Protocol	Length	Info
172.17.0.4	172.17.0.2	DNS	93	Standard query 0x78fe A mail.midominio.privado
172.17.0.2	172.17.0.4	DNS	142	Standard query response 0x78fe A 192.168.123.2

Figura 53: Consulta DNS 2 Wireshark quinta prueba

Como se puede apreciar, en este caso la relación es 93/142 por la anteriormente obtenida 91/121, siendo esta mayor. Por ello se decide probar de nuevo el ataque y observar si ese aumento en la amplificación ha sido significativo en lo que al desarrollo del ataque se refiere.

Nuevamente observamos como esto no es suficiente, y que, por tanto, se requeriría de un número mayor de atacantes para poder desarrollar con éxito el ataque.

7 Conclusiones

Para finalizar el estudio, se expondrán a continuación las conclusiones obtenidas derivadas de la experimentación y las pruebas realizadas. Pese a no haberse conseguido simular un ataque de denegación de servicio de manera satisfactoria, son muchas las conclusiones de valor que se han obtenido, y que hacen que la finalidad del estudio se vea cubierta, si bien es cierto que los objetivos de conseguir llevarlo a cabo y medir el tráfico necesario para ello, no se han podido cumplir.

En primer lugar, a lo largo del desarrollo del estudio, pudimos observar como el uso de un servidor DNS erróneo puede condicionar el resto de la investigación, obteniendo así conclusiones erróneas. Como se pudo comprobar, el uso de un servidor de tipo dnsmasq no es útil para llevar a cabo un ataque DDoS, ya que estos no eran capaces de cursar todo el tráfico entrante que les llegaba, haciendo de esta manera inútil cualquier intento de denegación de servicio. Este hecho no guardaba relación directa con el volumen de tráfico que tuviéramos, ya que su funcionamiento era similar independientemente del número de atacantes a los que atendiese, hecho que se verificó en el apartado 6.4.

Posteriormente en el apartado 6.5 se pudo verificar como con el uso de un servidor correcto, en este caso de tipo BIND, se obtenían unos mejores resultados, cursando todo el tráfico entrante y llevando a cabo de manera correcta la amplificación. Pero llegados a este punto, pese a encontrarnos usando el tipo correcto de servidor, aun así, no se logró llevar a cabo el ataque, lo cual deriva en la segunda conclusión obtenida.

Esta segunda conclusión, nos lleva al hecho de que para conseguir un ataque de denegación de servicio hemos de ser capaces de generar un volumen de tráfico tal que se consiga colapsar a la víctima y que esta no sea capaz de procesar las peticiones entrantes. Dicho así parece sencillo, pero tras llevar a cabo el estudio, se puede comprobar que no lo es tanto, en la última fase de la investigación, tres atacantes se encontraban generando grandes volúmenes tráfico dirigidos a una misma víctima, y pese a esto, no se logró la denegación de servicio. Este hecho nos lleva al punto de que se necesita de una gran botnet para poder llevar a cabo el ataque, pese a que este sea contra un único equipo, entendiendo por grande que el número de máquinas atacantes ha de ser mucho mayor que el de víctimas.

Como última conclusión y cierre del estudio, ésta ya derivada de todo el proceso de investigación y documentación previo, los ataques de denegación de servicio son un hecho cada vez más frecuente en nuestra sociedad, cada vez son más y de mayor duración y fortaleza. Por ello, se

ha de trabajar con la finalidad de desarrollar sistemas de seguridad que permitan bloquearlos de manera efectiva, evitando así que se conviertan en una práctica habitual ya no solo a grandes corporaciones, si no también a nivel de usuario.

8 English summary

8.1 Introduction

Nowadays, we are living a great worldwide technological revolution, our habits are experiencing a drift towards a complete connectivity with the world that surrounds us, facts that a few years ago seemed to be unthinkable, today are something common in our lives, a good example of it is the mobile phones introduction in our society, datum in which Spain is leader with an 88% of unique users. As datum of interest, in 2012, the smartphones introduction in Spain was about a 41%, in just five years, this value has been doubled reaching the 81%. Fact that reflects what previously has been said, that our society drifts towards a complete connectivity.

One aftermath of this growth in communication technologies is the rise of denial of service attacks, mostly known by its acronym DDoS. Numerous studies reflect the dramatic growth of this phenomena, nor only in the total number of attacks but also in its size and duration.

8.1.1 DoS and DDoS attacks. Concept

This section will address the definition of DoS (Denial of Service) and DDoS (Distributed Denial of Service), as well as the presentation of the three main attack types, without going into great detail in each one of them. Previously, before tackling this classification, clarify the pursued objective of these attacks, which is common in both cases (DoS and DDoS), to hamper the proper operation of the victim. Both DoS and DDoS can be divided in to three large groups:

- Volume Based Attacks: the attack's goal is to saturate the bandwidth of the attacked site, and magnitude is measured in bits per second (Bps).
- Protocol Attacks: this type of attack consumes actual server resources, or those of intermediate communication equipment, such as firewalls and load balancers, and is measured in packets per second (Pps).
- Application Layer Attacks: comprised of seemingly legitimate and innocent requests, the goal of these attacks is to crash the web server, and the magnitude is measured in requests per second (Rps).

8.1.2 Objective

As in any study or research, first of all we have to set some general objectives of large size, this way we can subsequently set specific goals that can be reached. The big picture of our objectives, was primarily delve into all related to denial of service attacks, and analyse both their ways of proceeding and the induced effect on the victims. As it has been seen in the previous

section, the field of knowledge covering such attacks is very wide and extensive, each attack requires a detailed study in order to know the true inner workings of these. Therefore, it was decided to establish as specific objective, the study and development of a volume based attack through DNS servers in a controlled environment. Three milestones where set to determine the development of the study: first of all, the creation of the controlled environment and the necessary elements; secondly, was to carry out the attack, testing different scenarios; thirdly and main one, the analysis of the obtained results and determine under which circumstances a denial of service attack is achieved.

8.2 Project history

8.2.1 Regulating frame

Firstly, we have to explain what cybersecurity means, and for its definition we resort to the 2.3 article of the Ministerial Decree 10/2013, 19 February where the Command Set of the Cyber Defence is created, published in the Official Bulletin of the Ministry of Defence. In this document, cybersecurity is defined as the set of activities aimed to protect the cyberspace against a wrongful use, defending its technological infrastructure, the services provided and the information handled.

In order to regulate everything related to cybersecurity and what this means, within BOE (Official Spanish Bulletin) is included in the electronic codes section, what is known as the Code of Law of Cybersecurity (updated on 9 March 2017). We can define cybercrime as any type of criminal activity that involves the use of computers or the internet. For the classification of this crimes we will use the one carried out by the Brigade of Technological Investigation of the Spanish National Police: attacks against the right to privacy; infringements of intellectual property through copyright protection; falsehoods; computer sabotage; computer fraud; threats; libel and slander; and child pornography.

Finally, the main bodies to take into account in terms of cybersecurity are: INCIBE, CERTSI, CNPIC and GDT.

8.2.2 Socioeconomic frame

The social and economic impact of the project is mainly due to the growth experienced by denial of service attacks in recent years. Such has been this rise, that they have come to affect our lives, either directly or indirectly.

Therefore, from de social aspect, the aim of the project is to understand how this attacks work, documenting it in detail so that the reader can understand how they are developed. This way, mixed with a more aware society in this regard, security measures could increase in personal devices, thus avoiding the easy creation of botnets that lead to a subsequent attack.

Regarding the economic impact, one of the main incentives of this attacks is the direct profit, by means of the ransom payment the victim avoids the attack, and therefore the economic losses this may cause. Finally, highlight the importance of the socioeconomic impact, as in a not-too-distant future, denial of service attacks could become an everyday item in our lives, and if we know how they work, we can minimize their consequences.

8.3 State of the art

8.3.1 DoS and DDoS attacks. Differences

- **DoS:** the main characteristic to take into account and the one that differences in its entirety from DDoS attacks, is that these attacks are carried out from a single computer, being its main objective, to flood the victim's machine through numerous requests, or directly, causing its collapse. Mainly they tend to be developed at local level, in LANs (Local Area Networks).
- **DDoS:** distributed denial of service attacks mainly differs from the already exposed DoS attacks, in the fact that, in this case, several joint machines are the ones that carry out the attack. It mainly seeks getting denial of service in resources or web servers, therefore connection to the internet is required in all machines involved.

  A series of steps have to be followed to carry out these attacks. Firstly, we have to take control over a computer, turning it into the main attack machine. Once we have control over it, it has to create a large botnet, group of computers that are controlled by the main one. This is primarily accomplished through the use of malware, or directly going through the authentication controls of the machine.

8.3.2 Tools

To carry out the study we made use of different tools, the ones we proceed to explain below.

- **Mininet:** it consists of a virtual machine image that gives us access to an operating system Linux Ubuntu, that creates a realistic virtual environment, with a real kernel, switches codes and applications that can be used freely.

- **Docker:** as its name suggests, this tool is based on the use of containers, this way we can run different applications simultaneously in isolated containers, so that we get better computing capacity in the machine. The use of these containers will be focus on the creation of the different elements required to carry out the study, such as the attackers, the victim and the DNS servers.
- **Hping3:** to carry out the attack, we require a tool that allows us to mask the source IP of the attacker and change it by the victim's one. To this end, we have hping3, which is a shell application that allows us to modify the source IP address for the one we want.
- **Wireshark:** it will allow us to analyse in real time and with great depth, everything that happens in our environment, so that we can easily check in the different interfaces, the source and destination IP addresses of the captured packages.
- **Htop:** it is a Linux real-time application which allows us to monitor those processes that are taking place in our machine. Thanks to it we can control the CPU and RAM memory consumption levels, which will let us know whether the attack is being effective or not.
- **Docker Stats:** as a complement to htop for those more complex cases in which we need to determine more precisely the percentage of CPU and RAM memory that is being used by each container.

8.4 Tests. Results

This section collects all the different scenarios that had been tested all over the study, starting with just one attacker and one DNS server, and progressively increasing the number of attackers, analysing the different results.

As it was expected, the first scenario gave us a negative result, with just one attacker we were so far from getting a denial of service in the victim, so we decided to increase this number. By the same token, in the second scenario again we were so far, having two attackers was not enough to carry out satisfactorily the attack, so again it was decided to add another attacker.

This new scenario had three attackers and just one DNS server, and the results obtained were again unsatisfactory. It was captured that the server could not process all the incoming packets, so that the outgoing traffic volume was negligible compared with the incoming one. So that, it was decided to reduce by one the number of attackers, and adding one more DNS server.

So that, now we have two attackers and two DNS servers, but again the servers show the same issue as in the previous scenario, they could not process all incoming traffic, fact that had no sense, as a DNS server is supposed to be able to process more traffic than the one they were

receiving. At this point, after testing different scenarios with a dnsmasq server which did not give us the expected results, we decided to change to a BIND DNS server, and check if at least it worked better than the previous one.

This final scenario consisted of three attackers and just one BIND server, and the results obtained were much better that in the previous ones. Despite we could not reach the denial of service in the victim, at least we could observe that the DNS server processed all the incoming traffic, making its amplification and sending it to the victim. This means that the attack was running properly, fact that after the previous four wrong scenarios is encouraging.

8.5 Conclusions

To complete the study, the findings derived from the experimentation and tests carried out will be posted below. Despite having failed to simulate a satisfactory denial of service attack, there are many conclusions of value that have been obtained, making satisfactory the purpose of the study.

Firstly, throughout the study we observed how the use of a wrong DNS server may condition the rest of the research, thus getting wrong conclusions. As it can be seen, the use of a dnsmasq server is not useful to carry out a denial of service attack, since it was not able to process all the incoming traffic, making useless any attempt of attack. Subsequently, when using a BIND server, the results obtained where much better, processing all the incoming traffic and carrying out properly the amplification of the packets. But despite using the correct server type, we failed to carry out the attack.

This derives into the second conclusion, that leads to the fact that to carry out the attack we have to generate a huge traffic volume, so that we can collapse the victim. In the last phase of the study, despite using three attackers this was not enough to carry out the attack, fact that leads to the point that a huge botnet is required, where the number of attackers must be much greater than the number of victims.

The third and final conclusion is obtained from the previous process of research and documentation, denial of service attacks are increasingly prevalent in our society, getting longer and stronger. Therefore, we have to work with the aim of developing security systems that allows to block them effectively, avoiding them to become a usual practice not only to big companies but also at user level.

9 Bibliografía

[1] Ditrendia-Informe Mobile en España y en el Mundo 2017. (2017). [PDF] Ditrendia. Available at: http://mktefa.ditrendia.es/informe-mobile-espac3b1a-mundo-2017

[2] Justo, D. (2017). *El uso de 'smartphones' en España se duplica en los últimos cinco años*. [online] Cadena SER. Available at: http://cadenaser.com/ser/2017/02/28/ciencia/1488281552_888684.html

[3] RedesZone. (2016). *Los ataques DDoS aumentan un 125% en comparación con el año pasado, según Akamai*. [online] Available at: https://www.redeszone.net/2016/06/11/los-ataques-ddos-aumentan-125-comparacion-ano-pasado-segun-akamai/

[4] Khalimonenko, A. and Kupreev, O. (2017). Los ataques DDoS en el primer trimestre de 2017. [online] Securelist - Información sobre Virus, Hackers y Spam. Available at: https://securelist.lat/ddos-attacks-in-q1-2017/84964/

[5] Khalimonenko, A., Kupreev, O. and Ibragimov, T. (2017). Los ataques DDoS en el segundo trimestre de 2017. [online] Securelist - Información sobre Virus, Hackers y Spam. Available at: https://securelist.lat/ddos-attacks-in-q2-2017/85272/

[6] Rivera, N. (2016). La diferencia entre un ataque DoS y un ataque DDoS. [online] Hipertextual. Available at: https://hipertextual.com/2016/09/ataque-ddos-dos-diferencias

[7] Incapsula.com. (n.d.). [online] Available at: https://www.incapsula.com/ddos/ddos-attacks/

[8] González, G. (2014). *Ataque DDoS: qué es y cómo funciona*. [online] Blogthinkbig.com. Available at: https://blogthinkbig.com/ataque-ddos

[9] Molinamateos.com. (n.d.). *Conceptos y definiciones | Ciberseguridad y Derecho*. [online] Available at: http://molinamateos.com/content/conceptos-y-definiciones-0

[10] MORENÉS EULATE, P. (2013). *Orden Ministerial 10/2013, de 19 de febrero, por la que se crea el Mando Conjunto de Ciberdefensa de las Fuerzas Armadas*. [ebook] Available at: http://www.emad.mde.es/Galerias/MOPS/novoperaciones/multimedia/documentos/20130226_CIBERDEFENSA.pdf

[11] BOE-173_Codigo_de_Derecho__de_la_Ciberseguridad.pdf. (2017). [ebook] Available at: https://www.boe.es/legislacion/codigos/codigo.php?id=173_Codigo_de_Derecho__de_la_Ciberseguridad

[12] Avast.com. (2015). Qué es el ciberdelito y cómo defenderse contra él | Avast. [online] Available at: https://www.avast.com/es-es/c-cybercrime

[13] Labs, D. (2015). Tipos de delitos informáticos – Delitos Informáticos. [online] Delitosinformaticos.info. Available at: http://www.delitosinformaticos.info/delitos_informaticos/tipos_delitos.html

[14] Policia.es. (n.d.). *Página oficial de la DGP-Comisaría General de Policía Judicial*. [online] Available at: https://www.policia.es/org_central/judicial/udef/bit_quienes_somos.html

[15] Web, S. (2017). Ciberdelitos: definición y tipos más frecuentes. [online] Seo-posicionamientoweb.com. Available at: http://www.seo-posicionamientoweb.com/ciberdelitos-definicion-tipos-frecuentes/

[16] Omicrono. (2016). Los 10 hackeos más grandes de Internet. [online] Available at: http://omicrono.elespanol.com/2016/09/hackeos-mas-grandes-de-internet/

[17] INCIBE. (2017). Qué es INCIBE. [online] Available at: https://www.incibe.es/que-es-incibe

[18] CERTSI. (2017). Qué es CERTSI. [online] Available at: https://www.certsi.es/sobre-certsi/que-es-certsi

[19] Cnpic.es. (2017). CNPIC. [online] Available at: http://www.cnpic.es/index.html

[20] Gdt.guardiacivil.es. (n.d.). GDT - Grupo de Delitos Telemáticos. [online] Available at: https://www.gdt.guardiacivil.es/webgdt/la_unidad.php

[21] Mediavida. (2011). ¿Cuanto se cobra por una hora de programación? [online] Available at: http://www.mediavida.com/foro/dev/cuanto-cobra-hora-programacion-431601

[22] autónomo, P. (n.d.). Precio de hora para consultor externo autónomo • Foros de SóloIngeniería.NET. [online] Soloingenieria.net. Available at: https://www.soloingenieria.net/foros/viewtopic.php?f=5&t=32065

[23] González, G. (2016). Varios ataques DDoS masivos afectan a grandes sitios como Twitter, Spotify y GitHub. [online] Genbeta.com. Available at: https://www.genbeta.com/actualidad/un-ataque-ddos-masivo-afecto-hoy-a-grandes-sitios-como-twitter-spotify-y-github

[24] ABC. (2017). Esto es lo que cobra un cibercriminal por un ataque DDoS. [online] Available at: http://www.abc.es/tecnologia/redes/abci-esto-cobra-hacker-ataque-ddos-201703242129_noticia.html

[25] Rouse, M. (2017). What is denial-of-service attack? - Definition from WhatIs.com. [online] Available at: http://searchsecurity.techtarget.com/definition/denial-of-service

[26] Paloaltonetworks.com. (2017). What is a Denial of Service Attack (DoS)? - Palo Alto Networks. [online] Available at: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

[27] Velasco, R. (2016). Ataques DoS y DDoS: Qué son y cómo podemos protegernos de ellos. [online] SoftZone. Available at: https://www.softzone.es/2016/07/14/ataques-dos-ddos-podemos-protegernos/

[28] Upload.wikimedia.org. (n.d.). [online] Available at: https://upload.wikimedia.org/wikipedia/commons/3/32/Tcp_normal_2.png

[29] Rouse, M. (2017). What is distributed denial of service (DDoS) attack? - Definition from WhatIs.com. [online] SearchSecurity. Available at: http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack

[30] Team, M. (2017). Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet. [online] Mininet.org. Available at: http://mininet.org/

[31] Docker. (2017). What is Docker. [online] Available at: https://www.docker.com/what-docker

[32] Docker. (2017). What is a Container. [online] Available at: https://www.docker.com/what-container

[33] RedesZone. (n.d.). Hping3: Manual de utilización de esta herramienta para manipular paquetes TCP/IP. [online] Available at: https://www.redeszone.net/gnu-linux/hping3-manual-de-utilizacion-de-esta-herramienta-para-manipular-paquetes-tcp-ip/

[34] Pavliksa, J. (2013). "Prueba de concepto" TA13-088A: DNS Amplification Attacks. [online] Viviendolared.blogspot.com.es. Available at: http://viviendolared.blogspot.com.es/2013/07/prueba-de-concepto-ta13-088a-dns.html

[35] Wireshark.org. (2017). Wireshark · Go Deep. [online] Available at: https://www.wireshark.org/

[36] Hipertextual. (2010). Comando Linux htop: administra interactivamente los procesos del sistema. [online] Available at: https://hipertextual.com/archivo/2010/03/comando-linux-htop-administra-interactivamente-los-procesos-del-sistema/

[37] Saive, R. (2017). Install Htop 2.0 - Linux Process Monitoring for RHEL, CentOS & Fedora. [online] Tecmint.com. Available at: https://www.tecmint.com/install-htop-linux-process-monitoring-for-rhel-centos-fedora/

[38] Docker Documentation. (2017). docker stats. [online] Available at: https://docs.docker.com/engine/reference/commandline/stats/

[39] Edrawsoft.com. (2017). All-In-One Cross-Platform Diagram Software for Flowchart, Org Chart and Mind Map. [online] Available at: https://www.edrawsoft.com/

[40] Support, T., Services, I. and Q&A, T. (2014). Network Address Translation (NAT) FAQ. [online] Cisco. Available at: https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html

[41] Naik, S. (2017). sameersbn/docker-bind. [online] GitHub. Available at: https://github.com/sameersbn/docker-bind

[42] Wiki.archlinux.org. (2017). dnsmasq (Español) - ArchWiki. [online] Available at: https://wiki.archlinux.org/index.php/Dnsmasq_(Español)

[43] Derussé, J. (2017). jderusse/docker-dns-gen. [online] GitHub. Available at: https://github.com/jderusse/docker-dns-gen

[44] Docker Documentation. (2017). Limit a container's resources. [online] Available at: https://docs.docker.com/engine/admin/resource_constraints/

[45] Stackoverflow.com. (2014). How to allocate 50% CPU resource to docker container? [online] Available at: https://stackoverflow.com/questions/26841846/how-to-allocate-50-cpu-resource-to-docker-container

[46] Naik, S. (2017). sameersbn/docker-ubuntu. [online] GitHub. Available at: https://github.com/sameersbn/docker-ubuntu

[47] pseudo-TTY, C. (2015). Confused about Docker -t option to Allocate a pseudo-TTY. [online] Stackoverflow.com. Available at: https://stackoverflow.com/questions/30137135/confused-about-docker-t-option-to-allocate-a-pseudo-tty

[48] Pavliksa, J. (2017). TA13-088A: DNS Amplification Attacks. [online] Viviendolared.blogspot.com.es. Available at: http://viviendolared.blogspot.com.es/2013/07/ta13-088a-dns-amplification-attacks.html